



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE INGENIERÍA

**Diseño y Construcción de un
Dispositivo Electrónico de Alerta
Remota para Situaciones de
Emergencia Vial**

TESIS

Que para obtener el título de
Ingeniero en Telecomunicaciones

P R E S E N T A N

Alejandro Fumero Rodríguez

Rolando Romero Feria

DIRECTOR DE TESIS

M.I. Juventino Cuéllar González



Ciudad Universitaria, Cd. Mx., 2016

Índice

GENERALIDADES	5
1.1 Objetivos Generales y Particulares.....	5
Objetivos Generales.....	5
Objetivos Particulares	5
1.2 Resumen	6
1.3 Introducción	7
1.4 Estado del Arte	10
Onstar	10
Detector Onbike (España).....	10
Phantom Tracking.....	11
Icedot	11
MARCO TEÓRICO.....	13
2.1 Microcontroladores	13
2.1.1 Introducción a los microcontroladores	13
2.1.2 Launchpad	17
2.1.3 Arduino UNO.....	19
2.1.4 Protocolos de comunicación serial	21
2.2 Sistemas de comunicaciones inalámbricas celulares.	27
2.2.1 Introducción a los sistemas de comunicaciones inalámbricas y móviles.....	27
2.2.2 Modelo de Interconexión de Sistemas Abiertos	30
2.2.3 Sistema Global para Comunicaciones Móviles (GSM).	33
2.2.4 Evolución del Sistema Global para Comunicaciones Móviles (GSM) a General Packet Radio Services (GPRS)	47
2.3 Sistemas de Navegación Satelital	50
2.3.1 Introducción a los Sistemas Globales de Navegación por Satélite (GNSS).....	50
2.3.2 Sistema de Posicionamiento Global (GPS).	55
2.3.3. Protocolo NMEA 0183	56
DESCRIPCIÓN Y DESARROLLO	59
3.1 Descripción del problema	59
3.2 Análisis del problema	66
3.2.1 Propuesta de solución al problema	66

GENERALIDADES

3.2.2 Decisiones de diseño	72
3.3 Implementación	82
3.3.1 Conexiones y descripción por etapas	82
3.3.2 Configuración de <i>software</i>	86
3.4 Pruebas de funcionamiento.....	92
3.4.1 Pruebas para el Arduino.....	92
3.4.2 Pruebas para módulo GPS	93
3.4.3 Pruebas para módulo GSM.....	94
PRUEBAS Y RESULTADOS	97
4.1 Resultados esperados	97
4.2 Resultados obtenidos.....	100
CONCLUSIONES.....	104
5.1 Conclusiones generales	104
5.2 Planes a futuro	106
REFERENCIAS BIBLIOGRÁFICAS.....	108
ANEXOS.....	112
Código Final	112
Desarrollo de Pruebas	118
Pruebas de precisión y exactitud del Módulo GPS	124
Glosario	126

DISEÑO Y CONSTRUCCIÓN DE UN DISPOSITIVO ELECTRÓNICO DE ALERTA
REMOTA PARA SITUACIONES DE EMERGENCIA VIAL

GENERALIDADES

1.1 Objetivos Generales y Particulares

Objetivos Generales

- Encontrar una solución para reducir el número de accidentes vehiculares fatales o que causen discapacidad.
- Brindar un mecanismo de aviso autónomo para vehículos en caso de requerir de los servicios de emergencia.
- Mejorar la perspectiva en cuanto a la seguridad de las motocicletas en la sociedad mexicana.
- Proponer un método para reducir el tiempo de atención a víctimas en caso de un incidente vehicular.

Objetivos Particulares

- Analizar de manera metódica el problema de los accidentes en motocicleta en México:
 - Estadísticas de accidentes fatales.
 - Características de una posible solución.
- Diseño y creación de una prueba de concepto como propuesta de solución:
 - Investigación de posibles componentes usados.
 - Interconexión de módulos y verificación de intercompatibilidad.
 - Creación de una secuencia de procesos, así como su diagrama de flujo.
 - Desarrollo e implementación de la secuencia de procesos en los dispositivos a usar.
 - Desarrollo de pruebas.

1.2 Resumen

Actualmente, en las ciudades con gran tránsito, como el Distrito Federal, Guadalajara y Monterrey, la motocicleta se ha convertido en un transporte alternativo, rápido y eficaz. Sin embargo, el uso de motocicletas no solo ha crecido en las grandes ciudades, sino en todo el país. A pesar de que resulta un excelente medio de transporte, el riesgo que representa viajar en una motocicleta en comparación a otros vehículos es más elevado.

Con este proyecto se pretende facilitar el acceso a servicios de emergencia en casos de accidentes o percances viales mediante el diseño y construcción de un dispositivo que permita comunicarse de manera automática y eficiente con un sistema de emergencias brindando datos relevantes del suceso y del usuario afectado.

Para la construcción del dispositivo se propone el uso de un microcontrolador conectado a un subsistema de geolocalización así como a un módulo de comunicación celular. Además, se deberán considerar sensores periféricos que permitan la detección de aquellos eventos para los que este dispositivo es diseñado.

Partiendo de que un motociclista ya cuenta con este dispositivo, en caso de ocurrir un accidente los sensores enviarán una señal al microcontrolador y este extraerá del módulo de geolocalización la posición actual, así como otros datos importantes. Estos datos serán procesados por dicho microcontrolador el cual se comunica de manera automática a través del módulo de comunicaciones celulares con un servicio de emergencias predeterminado.

1.3 Introducción

En México, el parque vehicular de motocicletas está cerca de los 2.5 millones¹, y algunos reportes estiman crecimientos anuales de hasta un 20 %. A pesar de ello, México aún se encuentra muy por debajo estadísticamente en comparación con otros países, tales como Brasil o los mercados europeos. Una de las mayores razones por la que la situación del mercado de motocicletas se encuentra en este estado es la preocupación al alto riesgo de conducir un vehículo de este tipo, particularmente en ciudades donde la percepción de poca educación vehicular es muy alta.

Según varios estudios, se estima que conducir motocicletas aumenta en 27 veces² la probabilidad de sufrir un accidente fatal en comparación con los automóviles, lo que hace que las preocupaciones previas tengan un fundamento sólido. A pesar de esto, las motocicletas representan una buena solución a los problemas de movilidad a los que se enfrentan las ciudades mexicanas, especialmente la megalópolis de la Ciudad de México.

Para reducir el número de accidentes fatales, existen varias etapas que podríamos clasificar en tres principalmente:

- Antes del accidente
- Durante el accidente
- Después del accidente

¹ <http://www.eluniversal.com.mx/articulo/autopistas/2015/12/22/crece-el-numero-de-motocicletas-y-de-robos>

² National Highway Traffic Safety Administration. 2016. Traffic safety facts, 2014: motorcycles. Report no. DOT HS-812-292. Washington, DC: US Department of Transportation.

Antes de que ocurra el accidente, las formas de evitar este son los métodos de prevención. Las leyes del tránsito, los chalecos de alta visibilidad, la conducción a velocidades moderadas y el entorno en donde se conduce la motocicleta son las formas más comunes para prevenir accidentes. Lamentablemente, el motociclista puede tener la responsabilidad de alguno de ellos, más no de todos. La educación vial del resto de la sociedad queda fuera del control del motociclista, y buena parte de los accidentes son causados por terceros.

En caso de que ocurra un accidente, el principal objetivo que se tiene en mente es conservar con vida al conductor y a los pasajeros. Para ello, existen formas de proteger las áreas del cuerpo más críticas o que resultan dañadas comúnmente, como cascos y armaduras o chamarras con protección. Incluso hay compañías que han buscado crear bolsas de aire para motocicletas, o evitar la caída del vehículo usando giroscopios. Este equipo especializado puede disminuir significativamente el riesgo de una fatalidad al momento de un accidente, pero lamentablemente no es infalible. El uso de este equipo cae completamente en la responsabilidad del motociclista.

Finalmente, en la etapa después del accidente se encuentran todos los cuidados de atención de emergencia, así como el tratamiento médico que el conductor o los pasajeros puedan recibir. Existen unidades especializadas en primeros auxilios para personas que han sufrido un accidente en motocicleta, así como tratamientos especiales en el caso de quemaduras sufridas por la fricción con el pavimento. En estos casos, el motociclista no puede llevar a cabo muchas acciones, ya que normalmente después de un incidente de este tipo, este queda incapacitado temporalmente.

Para el éxito de esta última etapa, el accidentado puede llegar a depender completamente de la acción de terceros, encargados de asegurarse que llegue la atención necesaria en el menor

GENERALIDADES

tiempo posible. Lamentablemente no siempre es así y la sociedad pocas veces asume responsabilidades en este tipo de situaciones.

Dada esta necesidad existente, surge la intención de crear un dispositivo que pueda detectar accidentes y proporcionar cierta información a los servicios de emergencia de manera automática, sin necesidad de intervención humana. Este dispositivo además proporcionaría un canal de comunicación a los accidentados, considerando que los medios propios del motociclista pudiesen resultar dañados debido al accidente.

Para este propósito se crea una prueba de concepto utilizando microcontroladores, módulos de comunicación GSM y un geolocalizador, además de simulaciones de sensores de impacto y un botón como medio de interacción con varias funciones del dispositivo.

1.4 Estado del Arte

En el mercado existen soluciones que atienden la problemática antes mencionada, por lo que se realizó una investigación de los principales sistemas disponibles.

Onstar

El sistema *Onstar* es una solución creada por el grupo *Chevrolet*, la cual busca ofrecer una solución integral de apoyo al conductor brindando asistencia vial, localización en caso de robo y detección de accidentes a través de la activación de las bolsas de aire. Esta solución se encuentra únicamente en vehículos de gama media alta y alta de la marca, por lo que no es posible instalarla en vehículos que no cuenten con este sistema instalado de fábrica. Sin embargo, muchas marcas están siguiendo los pasos de *Chevrolet* y comienzan a generar soluciones propias.

Detector Onbike (España)

Actualmente existe en España una solución implementada por el grupo *Detector*, una compañía que colabora con aseguradoras españolas, que inicialmente funcionaba como un sistema de localización y recuperación vehicular en caso de robo.

Para atraer al sector de los motociclistas, el grupo creó un sistema llamado *Onbike*, el cual busca brindar sistemas de localización de motocicletas, así como detectar sucesos tales como la caída o el robo del vehículo. En caso de que ocurra un accidente, el dispositivo lo detecta e intenta contactar al motociclista, y en caso de que no haya respuesta se contacta de manera automática a los servicios de emergencia

GENERALIDADES

Phantom Tracking

Phantom Tracking es una compañía norteamericana que, como producto principal, ofrece su tecnología patentada para la detección de accidentes en motocicleta. Además, ofrece un servicio de localización en caso de robo y de movimiento no autorizado. La manera en la que este dispositivo detecta las colisiones es a través de un acelerómetro sobre la motocicleta, el cual detecta patrones de velocidad sobre el vehículo, y al confirmar que la motocicleta se encuentra de lado, un operador intenta ponerse en contacto con el motociclista, y en caso de no recibir respuesta se pone en contacto con servicios de emergencia. El precio de este dispositivo es de 500.00 USD, y además el usuario debe pagar una suscripción de 15.00 USD al mes.

Icedot

Icedot es una empresa estadounidense que creó un sensor de colisiones para cascos, el cual se conecta a un teléfono inteligente a través de *Bluetooth*, y en caso de ocurrir un accidente utiliza el teléfono celular para enviar alertas a hasta 10 números celulares. Además, puede obtener la ubicación geográfica del módulo GPS del teléfono inteligente. El precio de este dispositivo es de 99.00 USD, además de necesitar una suscripción de 10.00 USD al año por persona.

Como podemos observar, el estado del arte actual nos muestra que existen soluciones diseñadas para resolver este problema proveniente de distintas partes del mundo. Sin embargo, estas soluciones aún están en una etapa de desarrollo, pues es difícil encontrarlas en el mercado, así como aplicarlas a vehículos de gama baja o motocicletas. *Onbike*, a pesar de compartir muchas funcionalidades con el proyecto a desarrollar, depende aún del factor humano, necesitando un operador para poder dar aviso a servicios de emergencia. Además, no es un dispositivo

únicamente dedicado a la parte de accidente, por lo que al incorporar otras funcionalidades puede elevarse el costo final para el usuario.

MARCO TEÓRICO

2.1 Microcontroladores

2.1.1 Introducción a los microcontroladores

Un microcontrolador es un chip altamente integrado que incluye la mayoría de las partes necesarias para formar un controlador embebido^{3*}. Típicamente incluye: CPU (Unidad Central de Procesamiento), RAM (Memoria de Acceso Aleatorio), EPROM (Memoria borrable programable de Solo Lectura), I/O (entrada y salida), temporizadores y controlador de interrupciones. Por lo general, los microcontroladores tienen un procesador de 8, 16 o 32 bits. El costo de los microcontroladores suele ser bajo (cerca de 2.88 dólares por unidad)⁴, y tienen funciones tales como manipulación de bits, acceso al I/O y procesos de interrupción.

Es importante hacer una distinción entre microcontrolador y microprocesador. Un microprocesador es un circuito integrado que incorpora todas las funciones de un CPU, mientras que un microcontrolador sí lo hace, tal como se ve en la figura 2.2.1. El microprocesador no incluye todas las partes que incluye un microcontrolador, por lo que se debe de considerar añadirlas en el diseño si se desea hacer uso de estas. Normalmente, el objetivo de los microprocesadores es crear un sistema de una microcomputadora, por lo que la capacidad de procesamiento es mucho mayor a la de un microcontrolador debido a que tienen el objetivo de

* Un controlador embebido es una computadora que es insertada en algún dispositivo cuyo propósito sea otro que el de proveer procesos de computación general.

⁴ <http://www.digikey.com/en/articles/techzone/2010/aug/microcontroller-prices-will-fall-despite-strong-demand>

procesar grandes volúmenes de datos, mientras que el microcontrolador se enfoca en operaciones de control de las entradas y salidas.

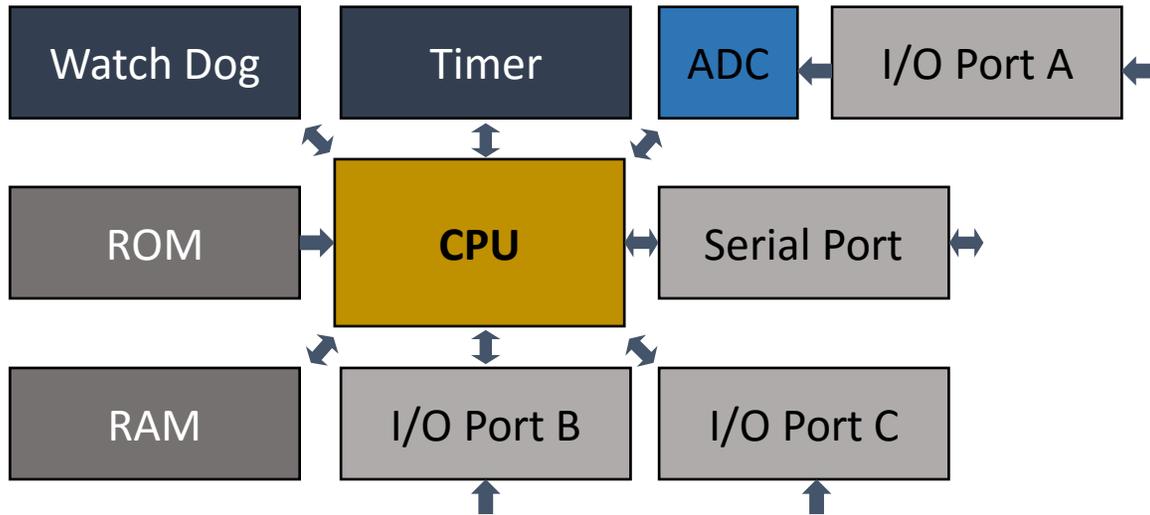


Fig. 2.1.1 Arquitectura de un microcontrolador típico.

El cerebro de un microcontrolador es su CPU, la cual tiene como funciones principales las operaciones de buscar y ejecutar instrucciones. Las instrucciones que el CPU puede entender están basadas en un conjunto de instrucciones predefinidas dependiendo de cada sistema. Para poder ejecutar una instrucción, el procesador debe buscar la instrucción en la memoria, decodificarla, ejecutarla y finalmente guardar el resultado en la memoria. A este proceso se le llama Ciclo de Máquina, y está en función de ciclos del reloj. Mientras menor sea la duración de un ciclo de reloj, es decir, que tenga una mayor frecuencia, menor será el ciclo de máquina, y por tanto se podrán ejecutar un conjunto de instrucciones en un tiempo más corto.

Dentro del CPU existen dos secciones importantes, la Unidad Lógica Aritmética (ALU por sus siglas en inglés) y los registros del CPU. En la ALU, como su nombre lo indica, se realizan todas las operaciones aritméticas y lógicas. Las operaciones básicas son la suma, la resta (usando complemento a 2), AND, OR y XOR. Todo lo que puede realizar el microcontrolador

MARCO TEÓRICO

tiene como base estas 5 operaciones. Los registros del CPU son los encargados de almacenar la información crítica para el funcionamiento del procesador, como el contador de programa, el apuntador del *stack*, datos a ser operados, etc. Estos registros son accesibles por medio de software, por lo que se pueden modificar dependiendo de la necesidad del programa.

La función de las memorias, tanto la RAM como la EPROM, es la de almacenar datos. Para lograrlo utilizan distintas formas de representar bits ya sea en estado “1” como en estado “0”.

Todas las memorias deben de cumplir dos propiedades básicas:

- La localización en donde cada dígito binario está almacenado debe ser direccionable de manera única.
- Debe ser posible leer el estado de cada dígito binario.

Dependiendo de si los datos siguen o no almacenados cuando el dispositivo se apaga, estas memorias pueden ser volátiles o no volátiles. La EPROM tiene memoria no volátil y la memoria RAM es volátil. Por lo tanto, la volatilidad es la propiedad de conservar el dato almacenado a pesar de que el dispositivo se encuentre apagado. Además de la volatilidad, una de las grandes diferencias entre ambas memorias es que acceder a la RAM es más rápido que acceder a la EPROM. Estas propiedades de la memoria son sumamente importantes pues hay que tomarlas en cuenta al momento de diseñar un sistema.

Normalmente, la memoria no volátil es utilizada para almacenar los datos de instrucciones del programa, así como datos que, al momento de iniciar el sistema, se deben definir para asegurar la correcta operación. En la memoria volátil se almacenan todos aquellos datos que durante el ciclo de vida del programa cambiarán frecuentemente.

El tamaño de una memoria también depende del tamaño del bus de dirección de memoria (MAB). Dependiendo del número de bits con el que se cuente para el MAB, será la capacidad

de mapeo de una memoria para el sistema. Por ejemplo, si se cuenta con un MAB de 16 bits, el espacio de la memoria sería de 64 kB⁽⁵⁾.

Los microcontroladores además cuentan con sistemas de interacción con el exterior, conocidos como puertos de entrada y salida. Estos puertos son la forma en la que se puede ingresar o extraer datos del sistema del microcontrolador. En muchos sistemas, estos puertos pueden ser usados tanto de entrada como de salida. Dependiendo del microcontrolador, también podrá existir la dualidad de usar tanto señales analógicas como señales digitales, estos puertos se conocerán como puertos de propósito general (GPIO por sus siglas en inglés).

La forma en la que el microcontrolador accede a estos puertos es a través de registros periféricos, a los cuales comúnmente se les conoce como entrada/salida mapeada en memoria. Esto facilita el uso de los puertos debido a que el acceso a ellos, tanto en lectura como escritura, es como leer una dirección en una memoria específica.

Para el correcto funcionamiento del sistema se requieren además módulos de reloj. Estos se componen generalmente de dos elementos básicos: el reloj y el temporizador. El reloj es un sistema que envía una señal cuadrada con un periodo específico de manera continua. Esto es el equivalente a un flujo constante de '1's y '0's digitales, permitiendo de esta manera que, a nivel físico, los transistores puedan cambiar de estado. Los módulos de temporizador son básicamente contadores que llevan la cuenta del número de ciclos de reloj que transcurrieron de un momento a otro.

Por último, los microcontroladores suelen contar con módulos tales como ADCs (convertidores analógico-digitales), comparadores y de comunicación digital. Los ADCs y los comparadores permiten al sistema trabajar con señales analógicas y poderlas traducir al sistema digital para

⁽⁵⁾ Esto se obtiene de la fórmula $2^{\text{Bits del MAB}}$

procesarlas, y el módulo de comunicación digital permite la transmisión y recepción asíncrona con lo que pueden intercambiar con sistemas externos datos que permitan ampliar las posibilidades de uso del sistema, o incluso comunicar dos microcontroladores para el intercambio de información.

2.1.2 Launchpad ⁽⁶⁾

Texas Instruments, en su afán de competir en el mercado de microcontroladores más amables para el público no especializado, el cual es encabezado por Arduino y sus chips Atmel, lanzó al mercado una línea de tarjetas de experimentación llamada Launchpad, la cual se caracteriza por su sencillo uso, su famoso modo de bajo consumo de energía y su completo software de desarrollo *Code Composer Studio*. A pesar de manejar varios chips en sus *launchpads*, nos enfocaremos en el MSP430, el cual es el chip más popular de esta compañía.

Texas Instruments es una compañía de electrónica que diseña y comercializa semiconductores. A la fecha, esta compañía es el tercer fabricante de semiconductores en el mundo. Fue fundada en 1951, y comenzó fabricando equipos para detectar sismos, así como electrónica de defensa militar. En 1954, fueron los primeros en fabricar un radio con transistores y son los responsables de la invención del circuito integrado. Durante las décadas de los 70's y 80's, se enfocaron en la electrónica de consumo. Actualmente tienen dos divisiones: Semiconductores y Tecnología Educativa.

En el año 2005 se crea Arduino y se genera una revolución en el mundo de la electrónica pues por primera vez se podía crear un proyecto de electrónica sin ser un experto en el tema y además

⁽⁶⁾ Para efectos de este trabajo, decidimos trabajar con circuitos de la compañía Texas Instruments, sin embargo, durante el desarrollo del proyecto se impuso la necesidad de cambiar este Launchpad por el Arduino Uno.

de manera accesible. Para poder competir en este espacio, Texas Instruments lanzó una línea de tarjetas de desarrollo, Launchpad, que tenían como componentes sus famosos chips.

Estas tarjetas tienen la ventaja de ser fácilmente programables, ya que cuentan con el programador dentro de la tarjeta, además de otros componentes básicos para su funcionamiento, tales como el cristal oscilador, capacitores, resistencias e incluso LEDs para probar ciertos programas. Además, se les puede dar poder a través de una conexión micro-USB, por

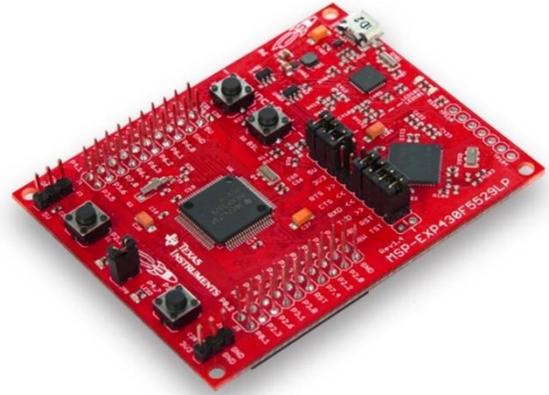


Fig. 2.2.1 Launchpad MSP430F5529

tanto su portabilidad y facilidad de encontrar fuentes de poder que se adapten a la tarjeta es realmente sencilla.

Los precios que se manejan para este tipo de tarjetas los hacen increíblemente accesibles, ya que no suelen rebasar los 25.00 USD. Esto hace que la creación de pruebas de concepto sea sumamente económica, y permite experimentar una variedad de combinaciones para encontrar la que más se adapte a nuestras necesidades.

Como se mencionó, el chip MSP430 es ideal para aplicaciones que requieran un consumo de energía muy bajo, pudiendo llegar hasta los $1.1 \mu\text{A}$ en un modo de ahorro. Cuenta con una arquitectura tipo RISC de 16 bits y un reloj de sistema de 25 MHz, así como con una memoria Flash de 128 kB. Cuenta además con 2 canales de comunicación UART, 16 canales ADC y 63 pines de salida.

Existen dos formas principales de programar este microcontrolador: utilizando el lenguaje ensamblador o utilizando el lenguaje C. Además, por esta visión de hacer mucho más sencilla

MARCO TEÓRICO

la adopción de estas tarjetas, se le han adaptado muchas de las librerías de Arduino, por lo que puede usarse un gran número de código fuente para esta plataforma.

2.1.3 Arduino UNO

Arduino es una placa de microcontrolador de bajo costo que comenzó como la colaboración entre estudiantes y profesores en el instituto IVREA, Italia, para crear microprocesadores más económicos que los comercialmente disponibles. Una de las grandes ventajas que tiene, además de ser *plug-and-play*, es que esta placa se puede conectar a todo tipo de sensores, motores, LEDs, y demás dispositivos y periféricos. Como ya se mencionó, Arduino constituyó una revolución en el mundo de la electrónica dada la facilidad para desarrollar proyectos sin ser un gran conocedor y por supuesto, dado su costo menor a los 30.00 USD. Además, Arduino se consolidó por ser de código abierto.

Como se comentó al inicio de esta sección, comenzamos usando el MSP430 de Texas Instruments, sin embargo, los módulos que se utilizan en el proyecto no son compatibles en cuanto al nivel de voltaje de trabajo se refiere, es decir, el TI trabaja a 3.3 V mientras que el Arduino UNO trabaja a 5 V, mismo voltaje al que trabajan los módulos SIM900 y GPS usados; incluso, en la fabricación de estos módulos, se considera la misma distribución de pines que presenta Arduino para sobreponer o conectar estos dispositivos uno sobre el otro y así evitar el uso de más cables.

El Arduino UNO es fácilmente programable y acepta diversos periféricos que funcionen con el protocolo de comunicación serial UART.

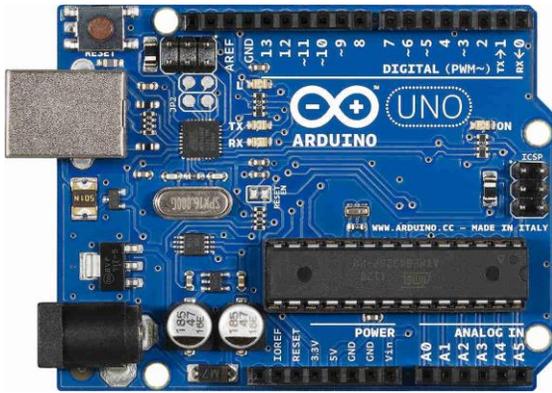


Fig. 2.1.3 Arduino UNO

En cuanto a conexiones, esta tarjeta de desarrollo cuenta con una entrada de 5 V de corriente directa así como una entrada serial que además de recibir y enviar datos de y a la computadora mediante el puerto USB, alimenta el dispositivo.

Este módulo de Arduino presenta 14 pines de entradas/salidas digitales, 6 entradas analógicas, un oscilador de cristal de cuarzo a 16 MHz, un botón físico de reinicio, y una memoria RAM estática o SRAM de 2 kB. En la figura 2.1.3 pueden observarse algunos de los aspectos mencionados.

Nuestro modelo presenta un microcontrolador Atmel ATMEGA328P. La programación de este debe ser a través del lenguaje propio basado en el lenguaje de programación de alto nivel *Processing*, similar a C++. Este microcontrolador tiene una memoria no volátil de 32 KB y un procesador con arquitectura de 8 bits. Esta tarjeta de desarrollo cuenta con un consumo de energía en activo de 0.2 mA. En modo de bajo consumo puede consumir 0.1 μ A y 0.75 μ A en un modo de ahorro de energía. La figura 2.1.4 muestra la distribución de pines del microcontrolador ATMEGA328P y sus funciones.

MARCO TEÓRICO

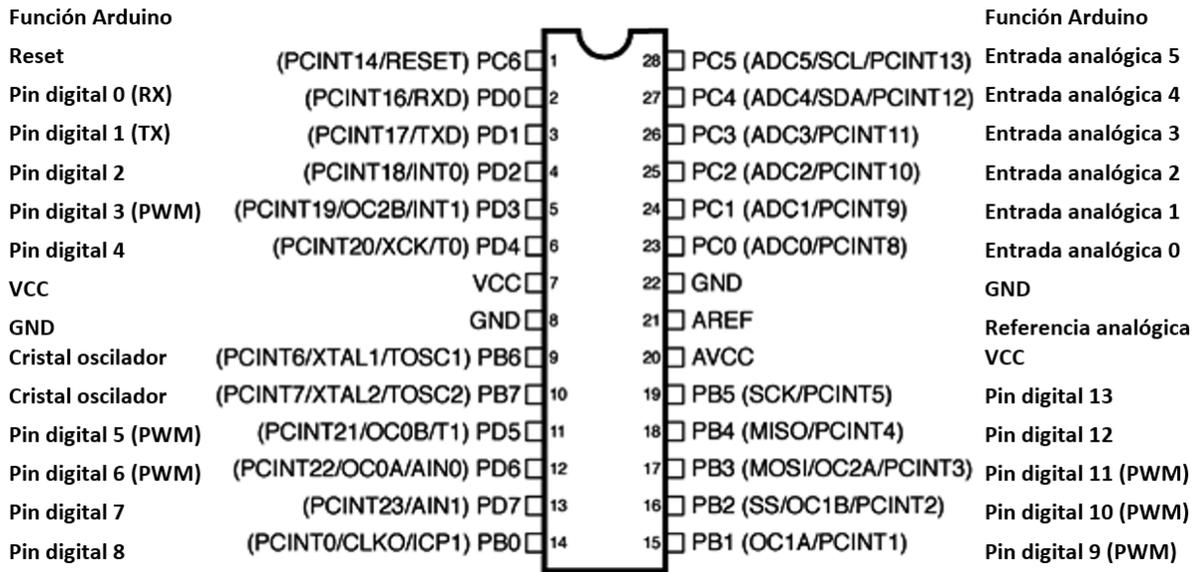


Fig. 2.1.4 Esquema pines para el ATMEGA328p

2.1.4 Protocolos de comunicación serial

Un microcontrolador necesita interactuar con sistemas externos a él. En el caso de que con quien necesite interactuar sea a su vez una computadora, procesador u otro microcontrolador, existen protocolos de comunicación que facilitan la transmisión y recepción de datos.

Existen dos formas básicas de comunicación entre los dispositivos: en paralelo y en serie. La comunicación en paralelo implica que a través de un bus de comunicación, que consiste de 8 o más líneas de transmisión, se transmita la palabra en bits de manera completa de un registro a otro. La comunicación serial consiste en transmitir un bit a la vez en una única línea de transmisión. Las ventajas de la comunicación en paralelo son muchas, pero debido a que por lo general se cuenta con un número de puertos I/O en los microcontroladores, comúnmente se opta por la comunicación serial. De esta manera, se sacrifica velocidad en la transmisión de datos a cambio de flexibilidad en el uso de puertos de entrada y salida.

Además, los protocolos de comunicación se pueden dividir en dos grandes familias: síncronos y asíncronos. La diferencia consiste en que la comunicación síncrona necesita una señal de reloj compartida entre los dos dispositivos, y la señal asíncrona elimina la necesidad de este tipo de señal. Para la transmisión síncrona, debido a que tanto transmisor como receptor comparten una referencia, se suele transmitir un bit por cada ciclo de reloj. Esto suele mejorar en gran medida la precisión en la recepción de los datos, así como la implementación en *hardware* del periférico de comunicación.

La transmisión asíncrona, por otra parte, no tiene un parámetro compartido para especificar cuándo inicia o cuándo termina un bit, por lo que los dispositivos deben establecer previamente la velocidad de transmisión de símbolos (conocido como *baud rate*). Por lo anterior, los dispositivos deben de contar con un reloj interno que esté cerca del *baud rate*, de lo contrario, comenzarían a ocurrir errores en la recepción. El *hardware* para implementar la comunicación asíncrona es normalmente más complicado que su opuesto, y se necesita usar, al menos, el primer bit y el último como parte de cada trama, lo cual reduce la tasa de transmisión real. Aunque no hay un límite teórico para la comunicación asíncrona, la mayoría de los dispositivos que trabajan de esta forma no suelen rebasar los 230 kbps.

Los microcontroladores generalmente vienen preparados para implementar varios de los protocolos de comunicación más comunes, por lo que no es necesario hacer mayor esfuerzo más que el de configurar los dispositivos que vayan a conectarse.

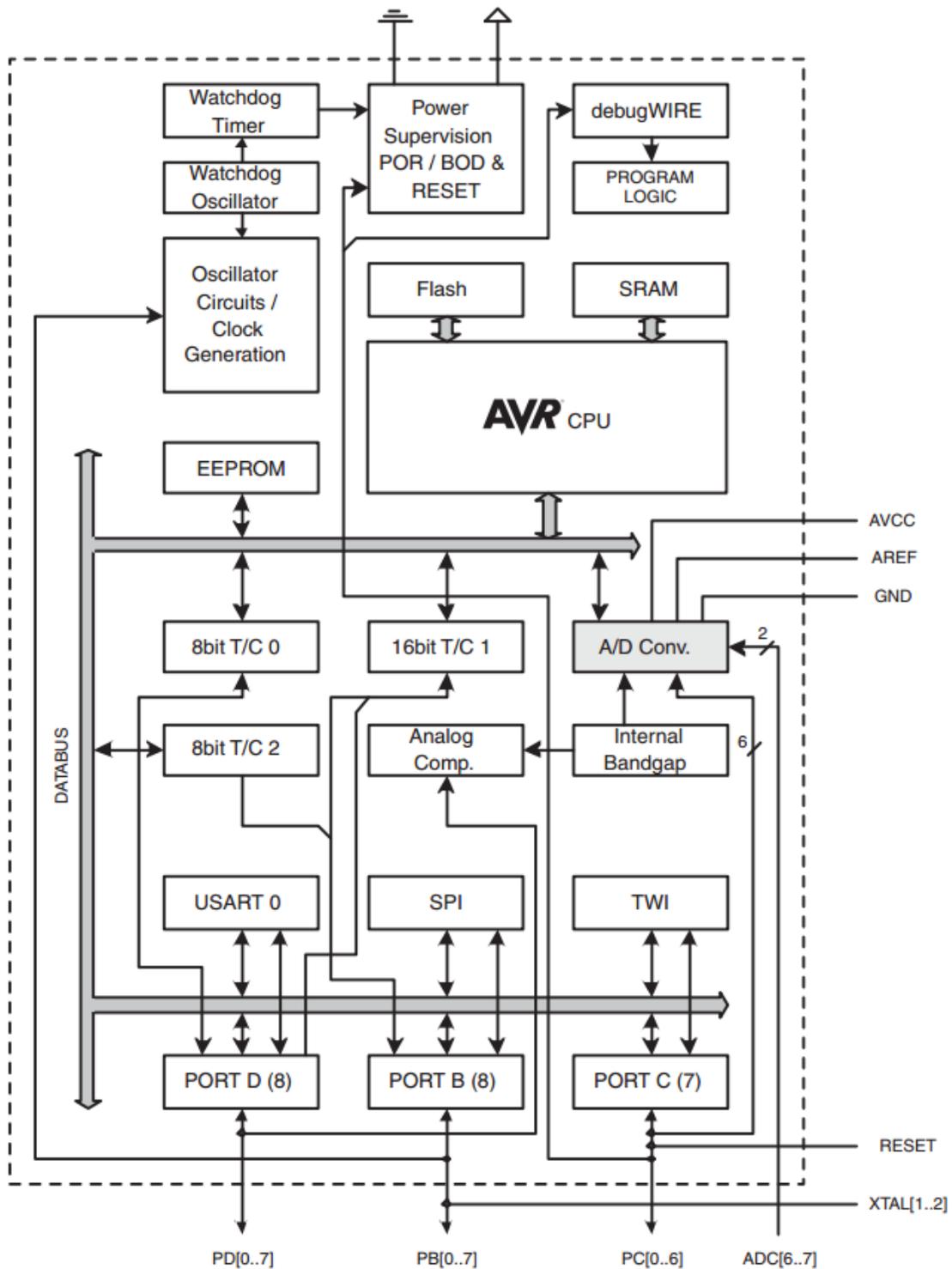


Fig. 2.1.5 Diagrama de bloques funcional del MSP430F5529, Datasheet de Texas Instruments

Los tres protocolos más importantes para los microcontroladores son el SPI, el I2C y el UART.

El UART (*Universal Asynchronous Receiver/Transmitter*) es el protocolo de comunicación serial más comúnmente encontrado en los microcontroladores. Consiste en sólo dos cables de transmisión entre los dispositivos, que iniciará en el puerto de transmisión del primer dispositivo y lo conectará con el puerto receptor del segundo dispositivo. Es un protocolo de comunicación *full-duplex*. Debido a que es una comunicación asíncrona, no es necesario que haya una línea de transmisión para compartir el reloj.

Este protocolo de comunicación utiliza otro protocolo de más bajo nivel, llamado RS-232, el cual establece los niveles necesarios de voltaje para que se pueda llevar a cabo la transmisión. El RS-232 establece que las señales deben tener un rango de -13 V a 13 V, y para este protocolo, una señal de voltaje bajo significa “1” lógico o bit de parada (*Stop Bit*) mientras que una señal voltaje alto significa “0” lógico o bit de inicio (*Start Bit*). Este protocolo comienza con el bit menos significativo (LSB) y finaliza con el bit más significativo (MSB) como se ve en la figura 2.1.6.

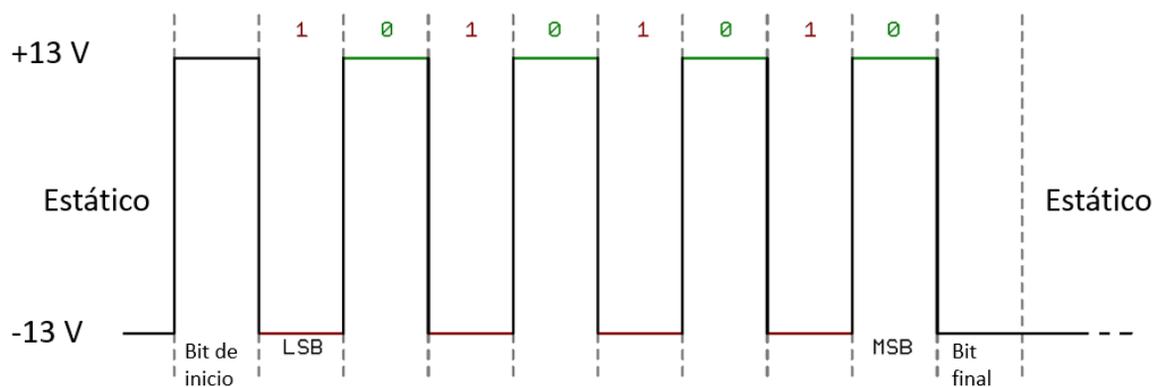


Fig.2.1.6 Valores lógicos del protocolo RS-232

Para poder enviar o recibir datos, se debe de pasar del estado de los bits en los registros (estado paralelo) al estado de los bits en el canal de comunicación (estado serial). Para ello

MARCO TEÓRICO

entra el circuito UART, que es el encargado principal de hacer la transformación serial-paralelo y viceversa. Además de cumplir con esta función y recibir y transmitir los bits, algunos circuitos UART contienen un *buffer*, el cual es un circuito de almacenamiento temporal de datos que permite guardar datos hasta que el microcontrolador sea capaz de procesarlos.

El SPI (*Serial Peripheral Interface*) es un protocolo de comunicación serial síncrono que se utiliza a distancias cortas. Este protocolo fue desarrollado por Motorola y adquirió un nivel de popularidad tal que se le considera un estándar. La arquitectura es de maestro-esclavo, y se puede comunicar en *full dúplex* como se observa en la figura 2.1.7.

A diferencia del protocolo UART, el SPI utiliza la comunicación serial síncrona, por lo que debe haber una tercera línea de transmisión en la que el reloj es compartido. El dispositivo que dicta la señal del reloj es conocido como el maestro y el dispositivo que la recibe es conocido como el esclavo.

Es posible que en ciertos dispositivos SPI exista una cuarta línea de transmisión o más, las cuales servirán como líneas seleccionadoras de esclavos del maestro, permitiendo así una arquitectura de múltiples esclavos.

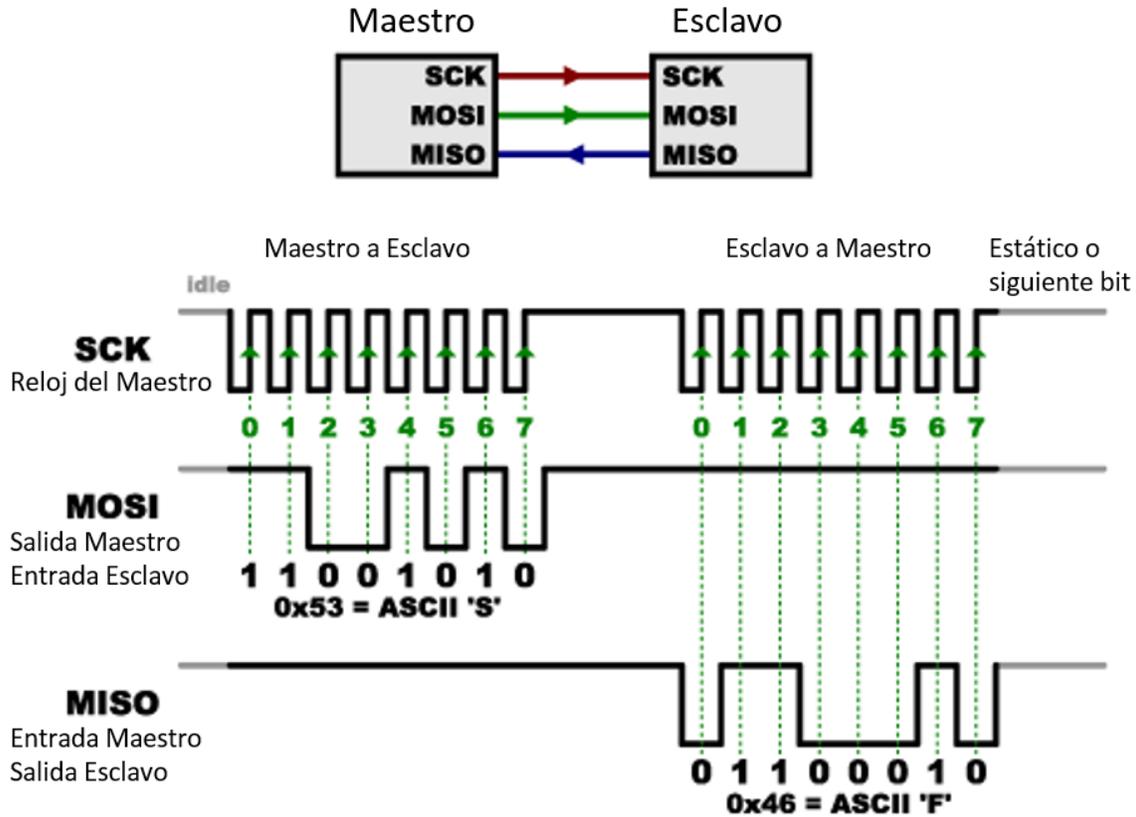


Fig. 2.1.7 Esquema de SPI con ejemplo de transmisión.

El I2C (*Inter-Integrated Circuit*) es también un protocolo de comunicación serial que maneja la arquitectura de maestro-esclavo y una comunicación *full-duplex*. La mayor diferencia entre el I2C y el SPI es que la primera es una comunicación asíncrona, es decir, que los dispositivos conectados no transmiten su reloj. Esto permite que se puedan implementar sistemas mucho más grandes que con SPI, e incluso implementar sistemas multimaestro.

2.2 Sistemas de comunicaciones inalámbricas celulares.

2.2.1 Introducción a los sistemas de comunicaciones inalámbricas y móviles

A pesar de que las primeras redes de comunicaciones móviles inalámbricas aparecieron en la década de 1940, la llamada Primera Generación o 1G de redes inalámbricas analógicas se concibió en los 70's en Estados Unidos. Esta red operaba con la transmisión de una imagen analógica a través del aire. El transmisor y el receptor se sintonizaban entonces en la misma frecuencia donde este último reconstruía y amplificaba la señal recibida.

Algunos de los inconvenientes de esta tecnología es la inexistencia de un *handover* (cambio automático del teléfono de la antena a la que está conectado a la nueva antena para transmitir y recibir) al cambiar uno o ambos usuarios del área de cobertura por lo que el usuario debía volver a conectarse a la red. Otro problema asociado a la 1G es la poca eficiencia en el uso del espectro puesto que era muy limitado el número de llamadas que se podían realizar simultáneamente.

La transmisión digital nació en los 80's en Europa donde se creó un nuevo estándar denominado *Groupe Spéciale Mobile* o GSM el cual posteriormente pasaría a llamarse *Global System for Mobile communications*, conocido en español como Sistema Global para Comunicaciones Móviles. Este nuevo estándar abrió las puertas a la denominada Segunda Generación o 2G. A partir de entonces se desarrollaron diversos sistemas inalámbricos destacando el Digital AMPS (D-AMPS), IS-95 (conocido también como *cdmaOne*), PDC (*Personal Digital Cellular*) en Japón y el ya mencionado GSM en Europa y Asia,

convirtiéndose este último en el sistema dominante gracias a la inclusión de servicios tales como el SMS (*Short Message Service*).

Además de lograr una considerable reducción en tamaño y peso en los dispositivos móviles gracias a la tecnología digital de los chips, la digitalización hizo posible que más usuarios pudieran usar el mismo ancho de banda incrementando así la eficiencia del sistema. Otras ventajas de esta nueva generación fueron la inclusión de correos de voz, llamada en espera y, como ya se mencionó, el SMS.

A pesar del gran desarrollo logrado por GSM, que para el año 2000 contaba con más de 400 millones de usuarios alrededor del mundo, se imponía la necesidad de que los sistemas soportasen aplicaciones para datos e Internet. Por este motivo se introdujeron los paquetes de datos o *packet data* a los sistemas GSM mediante el *General Packet Radio Services* (GPRS) o Servicio General de Paquetes vía Radio, como se conoce en español. Este nuevo sistema no es más que la adición de paquetes de datos al ya conocido GSM el cual continuaba manejando la voz de una manera eficiente.

Existen, entre otras, tres características que definen a los nuevos sistemas de paquetes de datos⁷:

- “Siempre conectado” donde se elimina el proceso de *dial-up*.
- Actualización de las redes existentes (GSM y TDMA) donde no era necesario cambiar equipos o modificar la infraestructura existente.
- Es una parte integral de los sistemas EDGE y WCDMA, dado que GPRS es la red núcleo de los paquetes de datos de estos sistemas 3G.

⁷ GPRS and 3G Wireless Applications; Anderson, Christoffer; Editorial John Wiley & Sons, Inc.; Primera Edición; 2002. Pág. 18.

MARCO TEÓRICO

Posteriores a este sistema GPRS aparecieron los comúnmente conocidos servicios de 3G y 4G. El primero de estos inició con el desarrollo de EDGE (*Enhanced Data rates for GSM Evolution*) el cual se basa en GPRS/GSM y se enfoca en mejorar la capacidad y la eficiencia de las comunicaciones al introducir un esquema de codificación más avanzado aumentando la cantidad de datos que pueden enviarse en cada *time slot*.

Otra tecnología que aprovechó las ventajas del 3G fue WCDMA (*Wideband Code Division Multiple Access*) o UMTS (*Universal Mobile Telecommunications System*) como se conoce en Europa.

En noviembre del año 2004 el proyecto de desarrollo de estándares 3GPP *Third-Generation Partnership Project*, comenzó sus estudios para determinar la evolución a largo plazo de su sistema UMTS. Esta investigación dio paso a la creación del estándar LTE o *Long Term Evolution* para comunicaciones inalámbricas de transmisión de datos de alta velocidad, el cual proporciona el punto de partida para una transición a 4G. Una diferencia significativa de esta nueva generación con respecto a la 3G, es que posee mayores tasas de transmisión y recepción de datos y además presenta retrasos más cortos en la transmisión.

Una de las características más notables de este sistema es la alta eficiencia espectral al utilizar la Multiplexación por División de Frecuencias Ortogonales (OFDM) como enlace descendiente lo cual ofrece una mayor robustez frente al multitrayecto, así como transmisión *broadcast/multicast* donde la misma información es transmitida desde múltiples estaciones base; y DFTS-OFDM como ascendente donde el modulador OFDM cuenta con un precodificador DFT. Otras características son: capacidad de trabajar con un ancho de banda adaptativo, es decir, anchos de banda de 1.4, 3, 5, 10, 15 y 20 MHz; transmisión desde múltiples antenas lo cual es una de las claves del gran desempeño del LTE ya que permite la multiplexación espacial o SU-MIMO donde diversos canales trabajan en paralelo.

Para anchos de banda de 20 MHz, *LTE release 1* ofrece enlaces de bajada y de subida de 100 Mbit/s y 50 Mbit/s respectivamente, esto para una conexión a dos antenas para bajada y a una para subida. En el caso de *LTE release 8* estas velocidades llegan hasta 300 Mbit/s y 75 Mbit/s respectivamente gracias al uso del multiplexado espacial de cuatro capas (4x4 MIMO), es decir, utiliza cuatro antenas en el enlace de bajada y modulación de amplitud en cuadratura 64QAM tanto para el enlace de bajada como para el de subida. Otra gran ventaja de esta tecnología LTE es que permite hasta 200 usuarios por celda donde esta puede tener 5 MHz.

Según el Sector de Normalización de las Telecomunicaciones (UIT-T), LTE es una red 3.9G en el estándar 3GPP porque no llega a los objetivos de la cuarta generación (4G) donde se requerían velocidades de 100 Mbit/s en movimiento y 1 Gbit/s en reposo para el enlace de descarga. Sin embargo, para el año 2010 la UIT declaró que aquellos proveedores con infraestructura LTE podían publicitarse como 4G. Ya en la actualidad todos los servicios de 4G se basan en la tecnología LTE gracias a la flexibilidad que ofrece esta.

2.2.2 Modelo de Interconexión de Sistemas Abiertos

A finales de la década de 1970 se impuso la necesidad de interconectar sistemas de comunicaciones de diferentes fabricantes por lo que la ISO (*International Standard Organization* u Organización Internacional de Normalización) decidió crear un nuevo subcomité (SC16) para el Modelo de Interconexión de Sistemas Abiertos u OSI por sus siglas en inglés (*Open Systems Interconnection*).

Este modelo OSI es una estructura que permite ver la red como la composición lógica de una sucesión de 7 capas, generalmente representadas de manera vertical, cada una con una tarea específica. En el nodo de una red, la capa de nivel N se comunica verticalmente con las capas

MARCO TEÓRICO

adyacentes $N - 1$ y $N + 1$ por medio de una interfaz que hace peticiones de servicio, donde esta capa de nivel N ofrece servicios a la capa situada por debajo.

Para la transmisión de mensajes de una capa a la otra, existe un proceso de encapsulado donde se agregan cabeceras y en ocasiones una cola. Para hacer referencia a los datos encapsulados que se intercambian entre las diferentes capas del modelo OSI se utilizan los llamados *Protocol Data Unit* o PDUs.

A continuación, se enumeran las 7 capas de este modelo OSI y se hace una breve descripción de su función.

Tabla 2.2.1 Capas del Modelo OSI

Capa	Descripción
7	Capa de aplicación. Este nivel proporciona una comunicación entre procesos o aplicaciones en distintos dispositivos, es decir, es la interfaz con el usuario.
6	Capa de presentación. Esta capa define el formato de los datos que se van a intercambiar entre las aplicaciones y ofrece a los programas de aplicación un conjunto de servicios de transformación de datos como la normalización, comprensión y encriptado de los mismos.
5	Capa de sesión. Gestiona el establecimiento de sesiones simultáneas de comunicación desde un mismo equipo, y se encarga de que el intercambio de información entre distintos usuarios se realice de forma sincronizada y organizada. Proporciona las estructuras de control para la comunicación entre aplicaciones.

- 4 Capa de transporte. Es la encargada de controlar el transporte de la información entre dos equipos, de forma que se asegure que la información transmitida llega al destinatario correctamente, así como de manera confiable.
- 3 Capa de red. Esta capa se enfoca en el direccionamiento y encaminamiento (o ruteo) de los datos en la red, es decir, se encarga de conectar equipos que están en diferentes redes ruteando paquetes del origen al destino.
- 2 Capa de enlace. Su principal función es la de ofrecer a la capa superior bloques de información sin errores. Por lo tanto, se encarga de realizar las operaciones necesarias que permitan controlar el flujo de bits por canal, de forma que un grupo de ellos (conocidos como tramas) sean facilitados al nivel superior sin errores, mediante la detección y corrección de aquellos que puedan llegar a producirse.
- 1 Capa física. Hace referencia al medio y a los equipos encargados de transmitir la información digital mediante las señales adecuadas. Es el único nivel que transporta realmente la información ya que el resto de las capas se ayuda de la inmediata anterior para realizar las operaciones de envío y recepción de datos.

En la práctica, estas comunicaciones no requieren del uso de todos los niveles antes descritos, sino que en la mayoría de los casos basta con tomar en cuenta los niveles inferiores del modelo, sobre todo los dos primeros: físico y enlace.

La Tabla 2.2.2 muestra, con referencia a cada capa, los diversos dispositivos y protocolos empleados por el modelo OSI.

Tabla 2.2.2 Dispositivos y Protocolos del Modelo OSI

Nombre de Capa	Protocolos y especificaciones	Dispositivos
Aplicación, presentación, sesión (Capas 5-7)	Telnet, HTTP, FTP, SMTP, POP3, VoIP, SNMP	Hosts, firewalls
Transporte (Capa 4)	TCP, UDP	Hosts, firewalls
Red (Capa 3)	IP	Router
Enlace de datos (Capa 2)	Ethernet (IEEE 802.3), HDLC	Switch LAN, puntos de acceso inalámbricos, modem de cable, modem DSL
Física (Capa 1)	RJ-45, Ethernet (IEEE 802.3)	Hub LAN, repetidor LAN, cables

2.2.3 Sistema Global para Comunicaciones Móviles (GSM).

A modo de antecedente, se muestra la Tabla 2.2.3 que recoge cronológicamente los principales sucesos en la evolución del GSM.

Tabla 2.2.3 Evolución del GSM

Año	Suceso
1982	La CEPT (<i>Conference of European Posts and Telecommunications</i>) establece el grupo GSM con el fin de desarrollar un sistema móvil celular europeo.
1985	Es aceptada una lista de recomendaciones creada por este grupo.
1986	Se llevan a cabo las primeras pruebas de campo para probar las diferentes técnicas de radio propuestas como interfaces aéreas.
1987	El <i>Memorandum of Understanding</i> (MoU) inicial es firmado por operadores de telecomunicaciones que representan a 12 países.
1989	La responsabilidad de las especificaciones del GSM pasa a manos del Instituto Europeo de Normalización de las Telecomunicaciones (ETSI, por sus siglas en inglés).
1990	Se entrega la primera fase de las especificaciones del GSM.
1991	Se lanza comercialmente el GSM.
1992	Aumenta la cantidad de países suscritos al primer MoU. La cobertura se expande a grandes ciudades y aeropuertos.
1995	Comienza la fase II de las especificaciones del GSM. La cobertura se expande a zonas rurales

La estructura celular

En un sistema celular, como es el caso de GSM, el área de cobertura se divide en células donde una célula es el área que un transmisor, o grupo pequeño de transmisores, puede abarcar. De esto se entiende que la potencia del transmisor determina el área de dicha célula.

MARCO TEÓRICO

La gran ventaja de este tipo de estructuras celulares es el uso de transmisores de bajas potencias con el fin de poder reutilizar las frecuencias.

La banda de frecuencias que se designa para un sistema celular se distribuye entre un grupo de células o *cluster* y esta distribución se repite en el área de cobertura del operador, donde las frecuencias usadas en una célula serán reutilizadas por diversas células separadas cierta distancia unas de las otras. Esta distancia evitará que exista interferencia en la comunicación. Como es evidente, el reuso de frecuencias incrementará considerablemente la capacidad de usuarios de dicho operador.

Para el correcto funcionamiento de estos sistemas se deben cumplir, entre otras, las siguientes dos condiciones:

- Las células vecinas no pueden tener la misma frecuencia. Para evitar las interferencias en estas, debe seguirse cierto patrón, por ejemplo, de siete células por *cluster*.
- La potencia de un transmisor dentro de una célula debe ser limitada con el fin de evitar las interferencias con transmisores vecinos.

La Figura 2.2.1 representa un sistema celular con un patrón de 7 células por *cluster*, esto se conoce comúnmente como $N = 7$. A pesar de ser este el sistema más extendido, existen también *clusters* de 4, 12, 21 o más células.

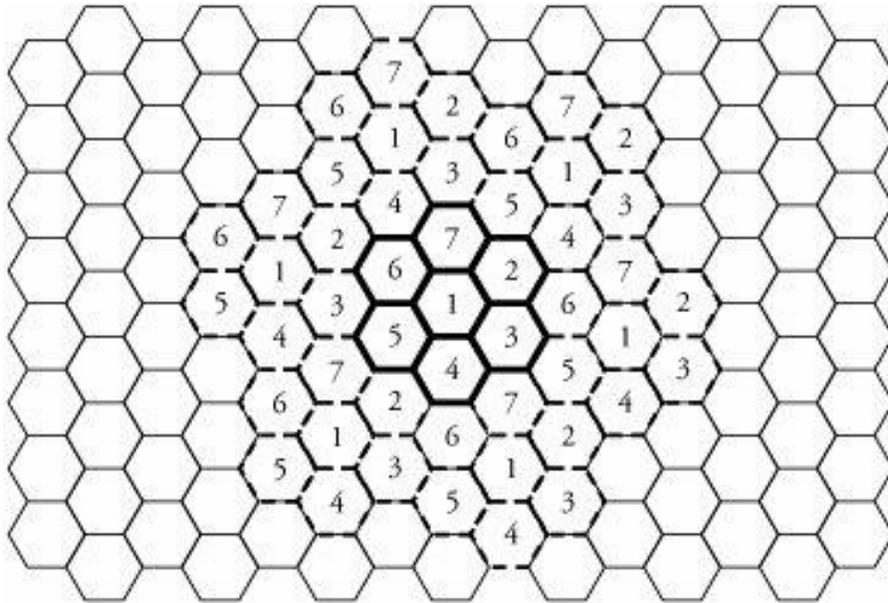


Fig. 2.2.1 Representación de un sistema celular ($N = 7$)

Servicios de telecomunicaciones

Usando las definiciones de la Unión Internacional de Telecomunicaciones, los servicios de Telecomunicaciones se dividen en:

- Teleservicios
- Servicios de portador
- Servicios suplementarios

Teleservicios

El servicio más básico de GSM es la telefonía, es decir la transmisión y recepción de voz. Además, cuenta con servicios de emergencia en donde el proveedor de servicios de emergencias más cercano es notificado mediante la marcación de tres dígitos.

Una característica única de los sistemas GSM en comparación con los sistemas analógicos más antiguos es el Servicio de Mensaje Corto (SMS). El SMS es un servicio bidireccional para enviar pequeños mensajes alfanuméricos de hasta 160 bytes. Para los mensajes punto a punto, un mensaje puede ser enviado a otro suscriptor del servicio y un mensaje de

MARCO TEÓRICO

confirmación de recepción se proporciona al remitente. El SMS también puede usarse para enviar mensajes tipo *broadcast*, es decir, de un usuario a muchos usuarios. Además, los mensajes SMS pueden almacenarse en el Módulo de Identidad del Suscriptor (SIM – *Subscriber Identity Module*). El servicio SMS resultó ser muy popular y para el año 2002 se habían registrado más de mil millones de mensajes enviados mensualmente.

En la fase II de GSM se introdujeron varias mejoras al sistema como por ejemplo, en los teleservicios, se introdujo una codificación de voz que redujo la velocidad de su transmisión de 13 kbps a 6.5 kbps, lo que permitió que los operadores ofrecieran su servicio al doble de usuarios sin tener que instalar una nueva infraestructura.

Servicios de portador

La naturaleza digital de GSM permite la transmisión y recepción de datos tanto síncrona como asíncronamente y pueden transportarse como un servicio de portador desde o hacia una terminal ISDN (*Integrated Services Digital Network*).

Los datos pueden usar un servicio transparente con un retraso fijo pero sin garantía de la integridad de los datos, o un servicio no transparente que garantiza la integridad de los datos a través de un mecanismo de petición automático de repetición (ARQ – *Automatic Repeat Request*) que desafortunadamente introduce retrasos variables. Las tasas de transmisión soportadas por GSM van desde los 300 bps hasta los 9600 bps, lo cual desde una perspectiva móvil eran suficientes para satisfacer las necesidades del momento.

Servicios Suplementarios

Los servicios suplementarios son características añadidas a la red celular que se instalan sobre los teleservicios o los servicios de portador. Algunos ejemplos de estos servicios son:

- Identificación de llamada.

- Llamadas multipartitas.
- Llamada en espera.
- Llamadas entrantes mientras se está en otra conversación.
- Grupo cerrado de usuarios.

Arquitectura de las redes GSM

Una red GSM se compone de diversos sistemas cada uno con una función específica. A continuación se hace mención de los mismos:

- Estación Móvil (*Mobile Station, MS*). Equipo terminal.
- Estación Base (*Base Transceiver Station, BTS*). Transmisor/receptor que enlaza estaciones móviles con la infraestructura fija de la red.
- Controlador de Estación Base (*Base Station Controller, BSC*). Administra un grupo de Estaciones Base.
- Subsistema de Estación Base (*Base Station Subsystem, BSS*). Conjunto de Estaciones Base y su BSC.
- Centro de Conmutación Móvil (*Mobile Switching Centre, MSC*). Proporciona el acceso hacia las redes telefónicas y RDSI (Red Digital de Servicios Integrados)
- Registro de Localización de Visitantes (*Visitor Location Register, VLR*). Base de datos para abonados visitantes en la red. Este registro en la red es temporal.
- Registro de Abonados Locales (*Home Location Register, HLR*). Base de datos para abonados locales.
- Centro de Autenticación de Abonados (*Authentication Centre, AUC*). Base de datos protegida donde se controlan los códigos confidenciales de los abonados.

MARCO TEÓRICO

- Subsistema de Conmutación de Red (*Network Switching Subsystem, NSS*). Conjunto conformado por: MSC, VLR, HLR, AUC. Es el componente que realiza las funciones de portar y administrar las comunicaciones entre teléfonos móviles y la Red Conmutada de Telefonía Pública (PSTN) para una red GSM.
- Centro de Control y Mantenimiento (*Operation and Maintenance Centre, OMC*). Garantiza la explotación técnica y comercial de la red.

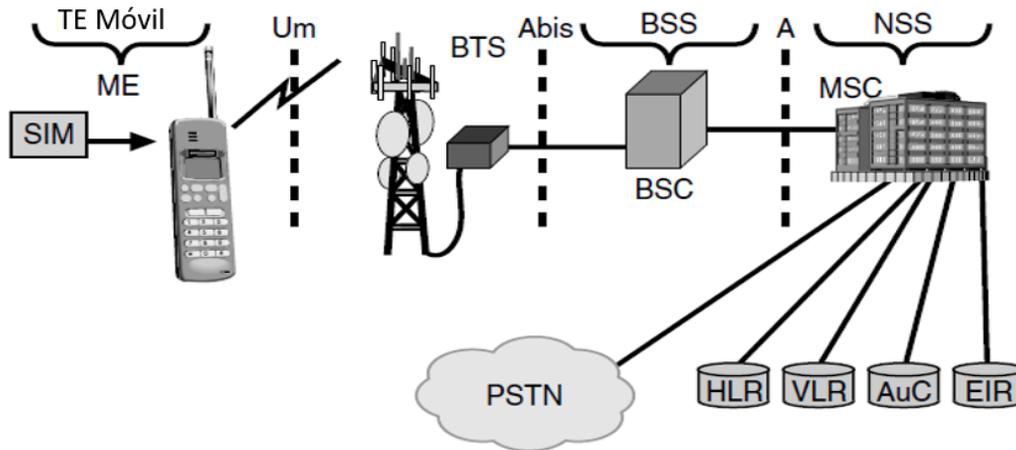


Fig. 2.2.2 Arquitectura de las redes GSM.

Fuente: GPRS. *General Packet Radio Service*; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 16.

Estación Móvil

La Estación Móvil consiste en el equipo físico, como el transmisor/receptor de radio, pantalla y procesador digital de señales y tarjeta SIM. Proporciona la interfaz aérea para el usuario de la red GSM. Además, proporciona servicios que incluyen teleservicios, servicios de portador y servicios suplementarios.

Módulo de Identificación del Suscriptor (SIM)

La tarjeta SIM provee movilidad personal para que el usuario pueda acceder a todos los servicios suscritos sin importar la locación de la terminal ni el uso de una terminal específica. Al insertar una tarjeta SIM en otro teléfono celular GSM, el usuario también puede recibir

llamadas en ese teléfono, hacer llamadas desde ese teléfono o hacer uso de otros servicios suscritos. Esta tarjeta contiene la identidad internacional móvil del suscriptor (IMSI – *International Mobile Subscriber Identity*) que identifica al suscriptor, una llave secreta de autenticación y otra información de usuario. Una contraseña o número de identificación personal pueden proteger a la tarjeta SIM contra un uso no autorizado.

Estación Base Transmisora/Receptora (BTS)

La Estación Base almacena los transmisores/receptores de radio que definen una célula y maneja los protocolos de enlace de radio con la Estación Móvil. En un área urbana relativamente grande pueden existir un gran número de BTS. Los requisitos para una BTS son:

- Robustez
- Confiabilidad
- Portabilidad
- Costo mínimo

Una BTS se conforma por los transmisores/receptores y las antenas usadas en cada una de las células de la red. Dependiendo de la cantidad de usuarios de la célula, una BTS puede tener desde 1 hasta 16 transmisores/receptores. Además, las BTS cumplen las siguientes funciones: codificación, encriptación, multiplexado, alimentación de la señal de RF a la antena, transcodificación y adaptación de la tasa de transmisión, y sincronización de tiempo y frecuencia, entre otros.

MARCO TEÓRICO

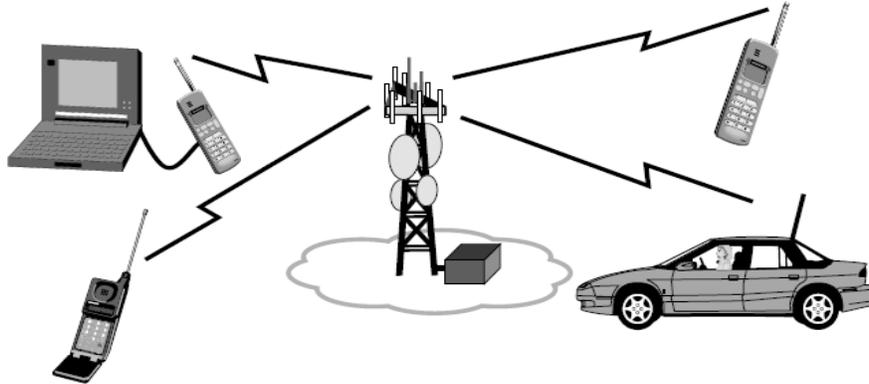


Fig. 2.2.3 Estación Base.

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 20.

Controlador de Estación Base (BSC)

El BSC administra los recursos de radio para una o más BTS. Maneja la configuración de canales de radio, saltos de frecuencia y *handovers*. El BSC es la conexión entre el teléfono móvil y el Centro de Conmutación Móvil. Este también traduce el canal de 13 kbps para voz usado en la conexión de radio al canal estándar de 64 kbps usado por la Red Conmutada de Telefonía Pública (PSTN). Funciones adicionales incluyen:

- Control de los saltos de frecuencia.
- Realizar la concentración de tráfico para reducir el número de líneas del MSC.
- Realocación de frecuencias.
- Sincronización de tiempo y frecuencia.
- Administración de la potencia.
- Mediciones de retardos de la señal recibida desde las estaciones móviles.

Subsistema de Estación Base (BSS)

Este subsistema se compone por dos partes, la BTS y el BSC, los cuales se comunican entre sí para permitir la operación entre componentes de distintos proveedores. Los componentes

de una BSS consisten de entre 4 y 9 células. Una línea independiente de alta velocidad (T1 o E1) se conecta desde la BSS hasta la Centro de Conmutación Móvil (MSC).

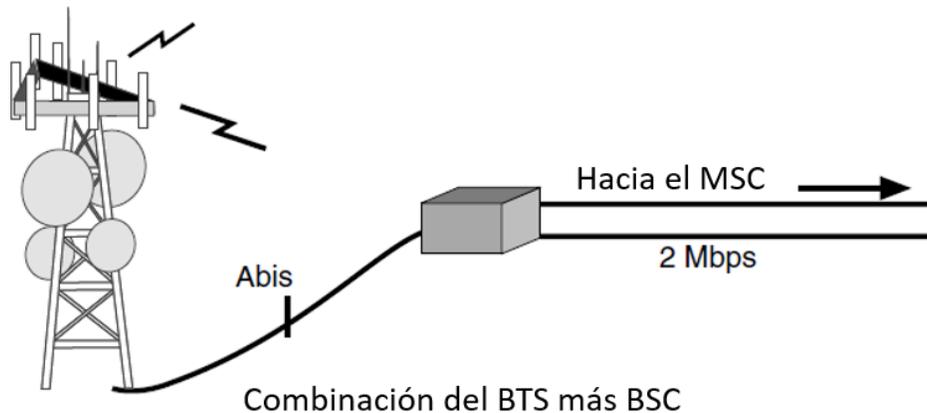


Fig. 2.2.4 Subsistema de Estación Base.

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 21.

Centro de Conmutación Móvil (MSC)

El componente central de un Subsistema de Red es el MSC. Este actúa como una Oficina Central (CO) de PSTN o ISDN, además de proveer toda la funcionalidad necesaria para manejar a un suscriptor móvil como registro, autenticación, actualización de locación, *handovers*, y enrutamiento de llamadas.

Las funciones principales de un MSC incluyen:

- Servicios de radiobúsqueda (*paging*).
- Coordinación de la configuración de llamadas para todos los MS del área operante.
- Alocación dinámica de recursos.
- Registro de locación.
- Funciones de trabajo interno.
- Administración de *handover*.
- Facturación.
- Encriptación.

MARCO TEÓRICO

Estos servicios son proporcionados en conjunto con varias entidades funcionales que forman el Subsistema de Red. El MSC provee la conexión a las redes fijas públicas y la señalización entre entidades funcionales usando el protocolo SS7 (*Signaling System number 7*), protocolo más común de las redes públicas.

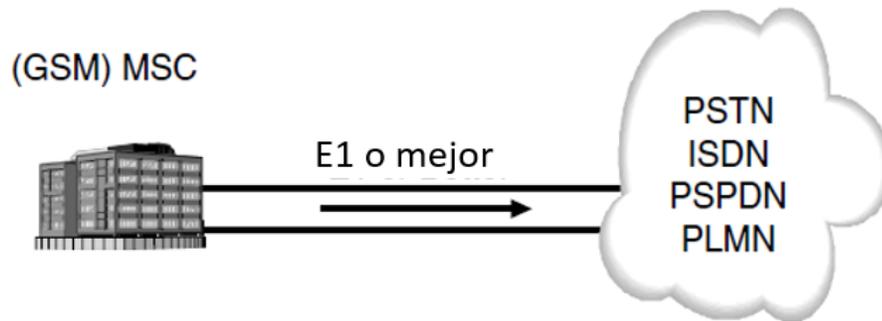


Fig. 2.2.5 Centro de Conmutación Móvil (MSC).

Fuente: GPRS. *General Packet Radio Service*; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 23.

Registro de Abonados Locales (HLR) y Registro de Localización de Visitantes (VLR)

Estos registros, junto con el MSC, proporcionan el enrutamiento de llamadas y las capacidades de *roaming* de GSM, llamados Subsistemas de Conmutación de Red (NSS).

El HLR contiene toda la información administrativa de cada suscriptor registrado en la red GSM correspondiente junto con la locación actual del móvil. Esta locación del móvil se presenta en forma de un Número de *Roaming* de Estación Móvil (MSRN) que es un número regular de ISDN usado para enrutar una llamada al MSC donde el móvil se encuentra.

El VLR contiene información administrativa selecta del HLR, que es necesaria para el control de llamadas y para proveer los servicios suscritos para cada uno de los móviles localizados actualmente en el área geográfica controlada por el VLR. Aunque cada identidad funcional puede implementarse como una unidad independiente, la mayoría de los fabricantes de

equipos de conmutación implementan un VLR junto a un MSC para que el área geográfica controlada por el MSC corresponda a aquella controlada por el VLR, simplificando la señalización necesaria.

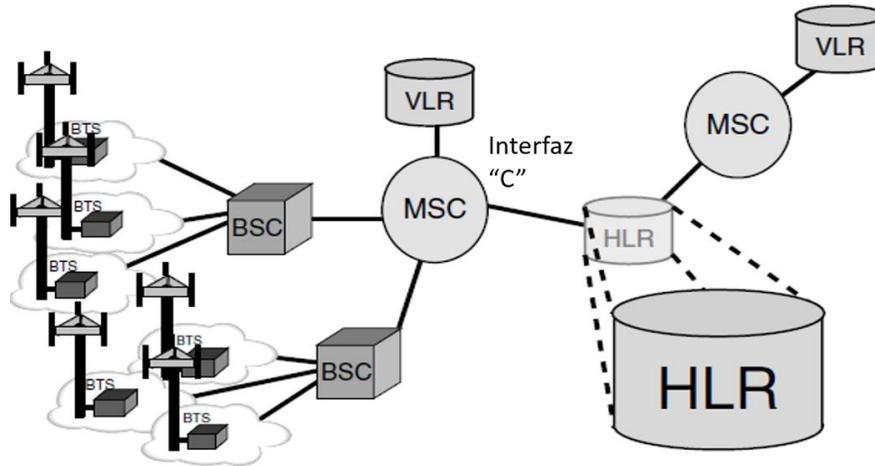


Fig. 2.2.6 HLR y VLR

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 25.

Red Pública Móvil Terrestre (PLMN)

La Red Pública Móvil Terrestre es el área en la que provee servicio un operador de red. Un país puede tener múltiples PLMN. Los enlaces entre una red GSM/PLMN y otras redes públicas pueden ser a nivel nacional o internacional. Todas las llamadas recibidas por una red GSM/PLMN se enrutan a un MSC como puerta de enlace (*gateway*). Todas las llamadas móviles se enrutan al MSC que funciona como puerta de enlace y todas las conexiones de llamada entre PLMN o redes públicas deben enrutarse a través de ciertos MSC designados.

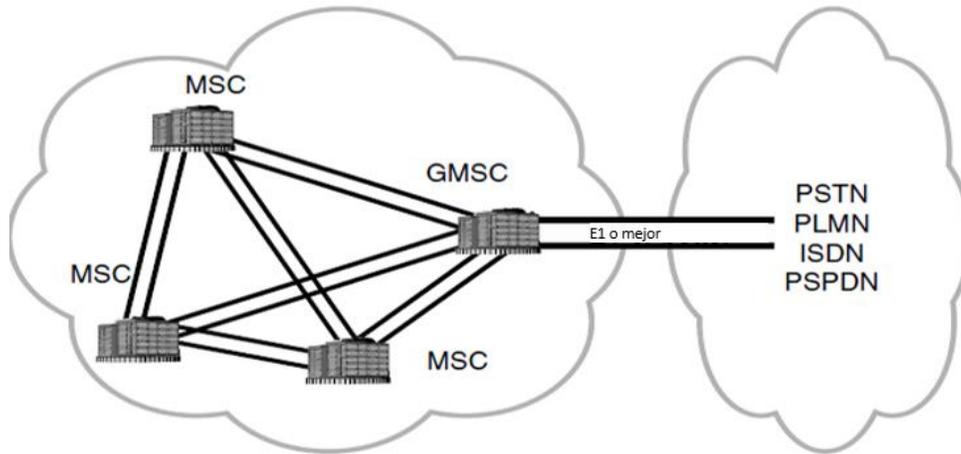


Fig. 2.2.7 PLMN

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 27.

Funcionamiento de las redes GSM bajo el modelo OSI

Por su naturaleza, las capas que más aplican para la arquitectura GSM y que satisfacen la mayoría de sus funciones, son las capas 1, 2 y 3.

Asegurar la transmisión de datos a determinada calidad a través del enlace de radio es solo parte de la función de una red móvil celular. Un móvil GSM puede trasladarse nacional e internacionalmente por lo requiere las funciones de registro, autenticación, ruteo de llamadas así como actualización de locación, las cuales existen y están estandarizadas para las redes GSM. Además, el hecho de que el área geográfica cubierta por la red esté dividida en células, implica la necesidad de implementar un mecanismo de *handover*. El Subsistema de Red realiza todas las funciones anteriores utilizando la Parte de Aplicación Móvil (MAP – *Mobile Application Part*) la cual está desarrollada sobre el SS7.

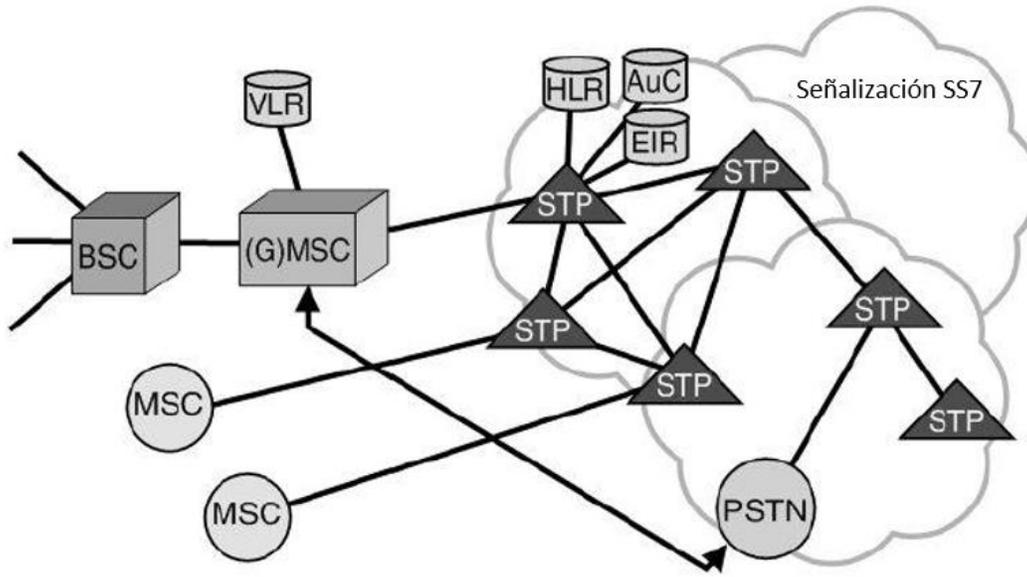


Fig. 2.2.8 SS7 y GSM trabajando en conjunto.

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 28.

Interfaces de la Red

Como expone Tisal en “La Red GSM”, las interfaces son una parte fundamental de la red. Estas “soportan el diálogo entre los equipos y permiten que funcionen entre sí. La normalización de las interfaces garantiza la correcta interacción entre equipos (...) producidos por distintos fabricantes.”

Tomando esto en cuenta, a continuación, se enlistan las interfaces normalizadas por la ETSI:

- Interfaz de radio Um. Interfaz más importante de la red localizada entre la MS y la BTS.
- Interfaz A-bis. Conecta una BTS con su BSC. Enlace por hilos MIC.
- Interfaz A. Localizada entre un BSC y un MSC. Enlace MIC a 64 kbps.
- Interfaz X.25. Conecta un BSC con el OMC. El soporte de enlace es proporcionado por una red de transmisión de datos.

MARCO TEÓRICO

- Interfaz entre el conmutador y la red pública. Definida por el protocolo de señalización 7 (SS7) del CCITT.

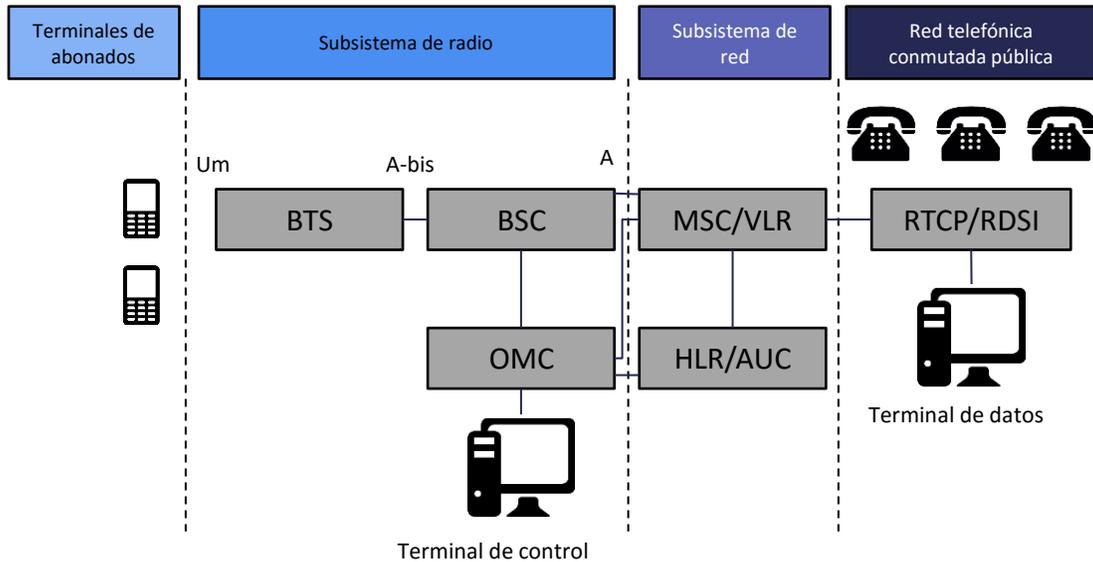


Fig. 2.2.9 Infraestructura y descripción de la red GSM

2.2.4 Evolución del Sistema Global para Comunicaciones Móviles (GSM) a General Packet Radio Services (GPRS)

La transmisión de datos en las redes GSM se vio superada por la necesidad de una mayor velocidad ya que la ofrecida por esta tecnología alcanzaba unos 9.6 kbps. Esto no se debía a un mal desempeño del Protocolo de Aplicaciones Inalámbricas (WAP por sus siglas en inglés) que se implementó en el año 2000 sobre la red GSM, sino a las redes de conmutación de circuitos, es decir, la transmisión de datos funcionaba como lo hacía una llamada telefónica regular donde el usuario marcaba a su ISP o proveedor de internet y se le asignaba un canal como los de radio pero para envío y recepción de datos a los ya mencionados 9.6 kbps sin poder compartir la capacidad del canal con nadie. Además de otros inconvenientes

tales como la irregularidad en la tasa de transmisión que no permanecía fija, WAP sobre GSM era muy costoso tanto para el cliente, que pagaba por el tiempo conectado y no por los datos transmitidos, como para el operador.

Para dar solución a estos problemas la ETSI recomendó la integración de las técnicas de transmisión “por paquetes”. Para ello, este organismo publicó especificaciones GSM Fase 2+ con el objetivo de introducir la tecnología GPRS a las redes GSM.

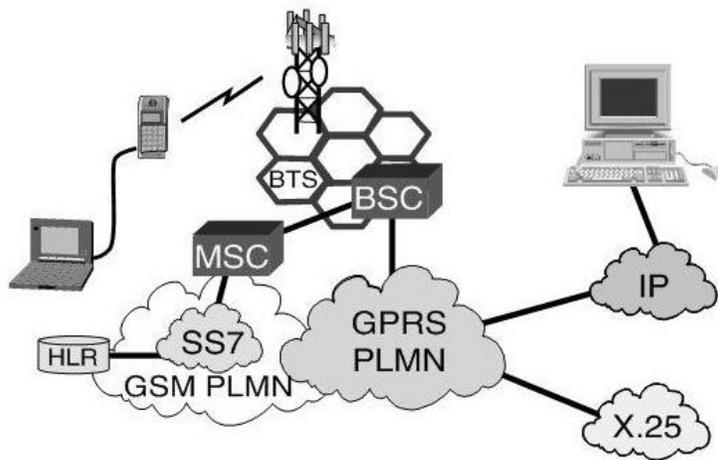


Fig. 2.2.10 Vista de la red GPRS

Fuente: GPRS. General Packet Radio Service; Bates, Regis J.; Ed. McGraw-Hill; 2002; Pág. 55.

El GPRS permitía entonces acceder a los servicios de internet a una velocidad de hasta 115 kbps que podían distribuirse en múltiples canales de radio pudiendo dedicarlos en su totalidad a un usuario o compartirlos con varios. Esto proporcionó una mayor flexibilidad y adaptabilidad del modo “paquete” con respecto al modo “circuito” logrando una mejor utilización de un sistema que ya existía (GSM) además de un mejor aprovechamiento de los recursos de radio que ahora se asignaban de manera dinámica.

Como se mencionó anteriormente, existen tres características principales que describen a los nuevos sistemas de paquetes de datos:

- “Siempre conectado”.

MARCO TEÓRICO

Mediante esta característica, GPRS permite al usuario estar siempre conectado u *online* sin necesariamente tener que pagar por minuto y además la conexión no es a través del proceso de *dial-up*.

- Actualización de las redes existentes (GSM y TDMA).

Gracias a que GPRS constituye una actualización de software para la mayoría de los equipos de los operadores, la compra e implementación de nuevos sistemas no fue necesaria. Además, esta migración no modificó la cobertura de los usuarios y permitió que los operadores acomodaran más usuarios en sus redes. Como muestra la Figura 2.2.11, el sistema GPRS se añadió al sistema GSM existente lo que ofrecía la posibilidad al usuario de no tener que cambiar su dispositivo móvil y además, no experimentar una degradación en el servicio.



Fig. 2.2.11 Paquetes de datos GPRS añadidos sobre las redes existentes

Algunas de las nuevas características o funciones que proporciona este sistema GPRS sobre GSM o TDMA son: chat; información visual y textual tal como clima, información de vuelos, horóscopos, noticias, entre otros; imágenes fijas o en movimiento como por ejemplo: fotografías, postales y presentaciones; búsqueda en la web mejorada gracias a mayores velocidades; trabajo colaborativo remoto al poder compartir y transferir archivos; correo electrónico corporativo y de internet; localización mediante GPS sin la necesidad de recurrir a servicios móviles de voz.

- Parte integral de los sistemas EDGE y WCDMA.

Como es lógico pensar, la evolución de estas tecnologías depende en gran parte de la urgencia por más velocidad de transmisión y recepción de datos, mayor capacidad así como la inclusión de nuevas características que permitan al usuario poseer múltiples herramientas en su dispositivo. Con la aparición del GPRS y los paquetes de datos, la migración o la evolución de los nuevos sistemas, tales como 3G y 4G, pudo realizarse de una manera más fácil y rápida. De esto se entiende que los sistemas 3G, por ejemplo, son una actualización de las redes GSM/GPRS, sin importar si estas son EDGE o WCDMA.

2.3 Sistemas de Navegación Satelital

2.3.1 Introducción a los Sistemas Globales de Navegación por Satélite (GNSS)

Un sistema global de navegación por satélite es una constelación de satélites que proveen un servicio de posicionamiento en la esfera terrestre. Se basa en la determinación por parte de un receptor satelital de su ubicación mediante la triangulación de la posición de por lo menos 4 satélites.

La humanidad siempre ha tenido la necesidad de ubicarse sobre la superficie del globo terrestre, y una de las formas más comunes era utilizar las estrellas como referencia. Esto se debe a que los cuerpos celestes se pueden observar desde cualquier parte del planeta y mantienen una posición relativamente conocida. Durante muchos años, esta forma de navegar y ubicarse en el planeta tierra continuaría siendo dominante.

Con el lanzamiento del primer satélite artificial, el *Sputnik I*, comenzó la carrera espacial. Ésta época se caracterizó por la búsqueda de explorar lo desconocido y colonizar el espacio.

MARCO TEÓRICO

Gracias a ella, comenzaron a surgir los sistemas de navegación por satélite como los conocemos hoy en día. El primero de ellos, el *Transit* de la década de los 60s, fue un sistema basado en el efecto *Doppler* que tenía una precisión alta. Posteriormente surgieron sistemas como el GPS estadounidense y el GLONASS ruso.

Actualmente, los sistemas de GNSS son vitales para muchas de las actividades humanas, desde el transporte aéreo hasta la navegación civil a través de un dispositivo de uso personal. Estos sistemas son tan importantes que actualmente existen, entre los sistemas americanos y rusos, más de 53 satélites operacionales, y países como la India y regiones como la Unión Europea se encuentran desarrollando sistemas independientes propios.

Funcionamiento

Debido a la complejidad de un GNSS, se debe de analizar su composición y funcionamiento para entender qué es lo que ofrecen este tipo de tecnologías y cómo podríamos explotarlos. Los sistemas globales de navegación se componen de tres partes principalmente: el segmento espacial, el segmento de control y el segmento de usuarios.

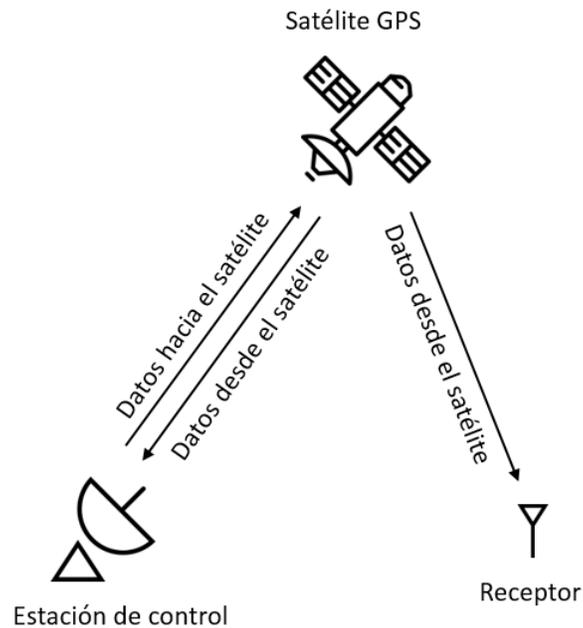


Fig. 2.2.12 Composición del sistema GNSS

Segmento espacial

Este segmento consiste en la constelación de satélites que transmiten señales de radio a los usuarios. Este tipo de satélites se ubican en la MEO (*Medium Earth Orbit* – Órbita Media de la Tierra) a una altura aproximada de 20 km.

A pesar de que el GNSS teóricamente sólo necesita 4 satélites para funcionar, para asegurar cobertura y disponibilidad de servicio mayor a 95 %, las constelaciones más importantes cuentan con por lo menos 24 satélites operacionales, e incluso con más para efectos de redundancia. Además, los satélites deben de estar colocados en planos orbitales lo suficientemente distintos para que se cubra toda la Tierra.

Esta es la parte más compleja de todo el sistema, y en donde la mayor parte de la inversión de estos sistemas termina.

Segmento de control

Este elemento es compuesto por una red global de estaciones terrenas encargadas de monitorear las transmisiones, dar seguimiento a los satélites, analizar y procesar datos y comandos de la constelación de satélites que componen el segmento espacial.

Segmento de usuarios

Como el nombre lo indica, este elemento es el que interactúa con los usuarios del sistema GNSS, ya sean de origen militar o civil. Este segmento está formado por todos aquellos dispositivos capaces de recibir y procesar las señales transmitidas por el segmento satelital.

Los receptores procesan la información y permiten al usuario conocer en tiempo real su latitud, longitud y altura, así como hora y velocidad. Las aplicaciones de estos datos son las que hacen que este segmento sea el más numeroso, pues existen muchos usos para esta información. Por ejemplo, los sistemas como bancos, mercados financieros y redes eléctricas dependen fuertemente de la sincronización temporal que este tipo de sistemas ofrece, y sería inimaginable pensar que operarían sin ellos.

Cálculo de posición

La forma de operar de un sistema de navegación por satélite se basa en principios muy sencillos. La forma en la que podemos conocer nuestra posición si conocemos la posición y distancia de los satélites es conocida como trilateración esférica. De manera sencilla, si sabemos que nuestra posición se encuentra a una cierta distancia de cuatro puntos en el espacio, y generamos esferas con estos radios, nuestra posición será la única intersección de las cuatro superficies.

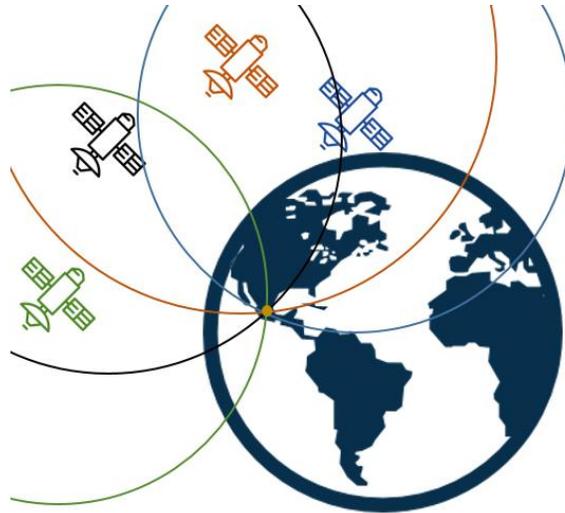


Fig. 2.2.13 Cálculo de posición

Para poder realizar la trilateración ciertos datos son necesarios. El primero de ellos, la posición del satélite, es relativamente sencilla de conocer dado que los satélites de las constelaciones de los sistemas GNSS viajan en órbitas fijas y predecibles. Los satélites se encargan de enviar su posición actual para que los receptores cuenten con esta información sin mayor problema.

El segundo dato necesario para conocer la posición es la distancia que hay entre los satélites y el receptor. Determinar esta distancia es un proceso complejo, ya que es necesario que el receptor realice cálculos. Dado que los satélites transmiten señales de radio, las cuales viajan a la velocidad de la luz, los dispositivos únicamente necesitan conocer una diferencia de tiempo para ser capaces de calcular la distancia.

El método para conocer esta diferencia de tiempo es mediante el envío de código pseudoaleatorio por parte del satélite. Este código es generado utilizando el reloj atómico con el que cuentan los satélites de este servicio, mismo que los receptores terrestres pueden generar con los relojes de cristal de cuarzo. Estos códigos son los mismos en tiempos iguales, sin importar con qué tipo de reloj sean generados. Al conocer el código enviado por el satélite,

MARCO TEÓRICO

el receptor puede calcular el tiempo transcurrido, obteniendo posteriormente la distancia al satélite. Una vez que se cuente con las cuatro posiciones satelitales y sus respectivas distancias, el receptor puede obtener la latitud, la longitud y la altura a la que se encuentra.

Una vez conocida la posición del receptor y sincronizado su reloj, el proceso puede reiniciarse. Si se le permite, el dispositivo del usuario puede mantener un registro de las posiciones y los tiempos en su memoria, pudiendo así conocer información útil tal como la velocidad promedio de movimiento, la ruta que se ha tenido, y el tiempo aproximado de llegada si se trazó un camino a un destino.

2.3.2 Sistema de Posicionamiento Global (GPS).

Como puede observarse, la información obtenida por un sistema GNSS es de mucha utilidad. El gobierno americano desde la década de los 60s se dio cuenta del potencial que un sistema GNSS podría brindar, y durante la carrera espacial comenzó a experimentar con sistemas que después evolucionaron a los conocidos actualmente.

El sistema *Transit*, el cual se basaba en el efecto *Doppler* para poder calcular la distancia entre el receptor y el satélite transmisor, fue implementado por la marina norteamericana y consistía de 6 satélites que tenían como objetivo la localización de submarinos. El sistema *Transit* no era muy eficiente pues los submarinos tenían que esperar varias horas para poder recibir las señales, pero sentó las bases del sistema GPS.

Después de 11 años de investigación y trabajo, en 1974 se lanza el primer satélite del sistema *Navstar*, que después se convertiría en el sistema GPS. Este lanzamiento marcó el inicio de una etapa de pruebas en las que se lanzaron once satélites más. En 1983, después de un incidente con un vuelo coreano, el presidente Ronald Reagan ofreció el permiso de uso del

sistema GPS, una vez que fuera terminado, a todos los vuelos comerciales civiles, lo que cambió la visión de un sistema GNSS con fines únicamente militares.

En 1989 se envió el primer satélite GPS completamente operacional al espacio, concluyendo el envío de los 27 satélites que conformaban la primera generación del sistema GPS en 1995.

No fue hasta 1998 que Al Gore anunció que habría un plan para hacer que los satélites transmitieran señales distintas, con el fin de que las aplicaciones civiles pudiesen existir sin interferir con el uso militar. En los inicios del nuevo milenio, los dispositivos personales civiles que empleaban esta tecnología comenzaron a proliferar. Actualmente, y después de evoluciones al sistema, Estados Unidos cuenta con una constelación de 31 satélites operacionales, y el sistema GPS III tiene planes de iniciar sus lanzamientos en 2016.

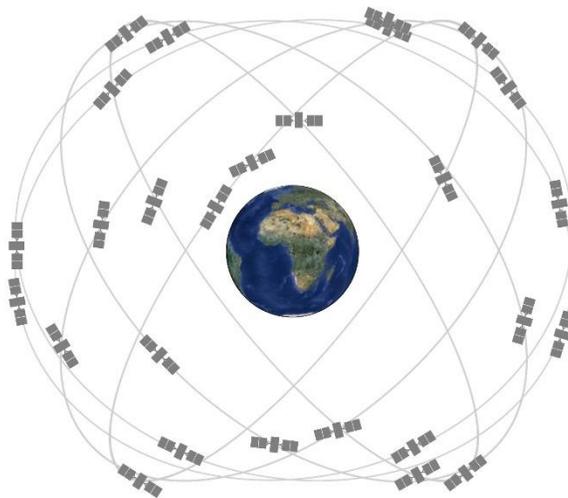


Fig. 2.2.14 Sistema de Posicionamiento Global

2.3.3. Protocolo NMEA 0183

El protocolo NMEA (*National Marine Electronics Association*) es una especificación eléctrica y de datos para el funcionamiento de sistemas electrónicos marinos, tales como sonares autopilotos y GPS. El protocolo utiliza ASCII y estándares seriales muy simples que

MARCO TEÓRICO

ayudan a comunicar a los dispositivos haciendo de la interpretación de los mensajes una tarea mucho más fácil de realizar.

Este protocolo fue realizado por NMEA, la cual es una asociación sin fines de lucro que reúne a fabricantes, distribuidores e instituciones de educación que comparten el interés por la comunicación entre instrumentación marina. Este estándar voluntario fue lanzado en marzo de 1983 y desde entonces ha sido revisado y actualizado. A pesar de esto, el protocolo NMEA 0183 busca ser sustituido por el protocolo NMEA 2000, pero un número considerable de dispositivos aún maneja el protocolo anterior.

Este estándar funciona a 4800 bauds, en un bus de datos serial que maneja 8 bits.

El formato de las oraciones que estandariza este protocolo es como sigue:

$$\$ttsss,d1,d2,\dots <CR><LF>$$

Como se puede observar, las oraciones inician con el símbolo \$, seguido de identificadores del dispositivo (*tt*). Para nuestro dispositivo, estos dos símbolos siempre serán GP (*Global Positioning*). Después de esto, siguen los identificadores de oración (*sss*), seguido de los campos de datos, que son separados por una coma. Al final de la oración, como datos opcionales, se pueden encontrar un *checksum* y un indicador de línea final.

La oración que más se utiliza en el proyecto es la de Datos de Fijación de Sistema de posicionamiento global, que tiene el siguiente formato, así como los siguientes campos:

$$\begin{array}{cccccccccc} \$-RMA,A,lll.ll,a,yyyyy.yy,a,x.x,x.x,x.x,x.x,x.x,a*hh \\ | \quad | \\ 1 \quad 2 \quad 3 \quad 4 \quad 5 \quad 6 \quad 7 \quad 8 \quad 9 \quad 10 \end{array}$$

- 1) Aviso de parpadeo
- 2) Latitud
- 3) N o S (Norte o Sur)

- 4) Longitud
- 5) E o W (Este u Oeste)
- 6) Diferencia de tiempo A
- 7) Diferencia de tiempo B
- 8) Velocidad sobre tierra
- 9) Datos de control adicionales
- 10) *Checksum*

De estos, los datos más utilizados son los campos 2, 3, 4, 5 y 8

DESCRIPCIÓN Y DESARROLLO

3.1 Descripción del problema

El presente proyecto propone la construcción de un sistema electrónico enfocado en la comunicación vía SMS de un usuario con los servicios de emergencia en casos de accidentes viales, específicamente en accidentes que involucran motocicletas, aunque su uso puede extenderse a vehículos motorizados de otros tipos tales como ómnibus o automóviles.

Actualmente en el mundo más de 1.2⁸ millones de personas pierden la vida en accidentes de tránsito y entre 20 y 50 millones sufren heridas y traumas relacionados con estos accidentes.

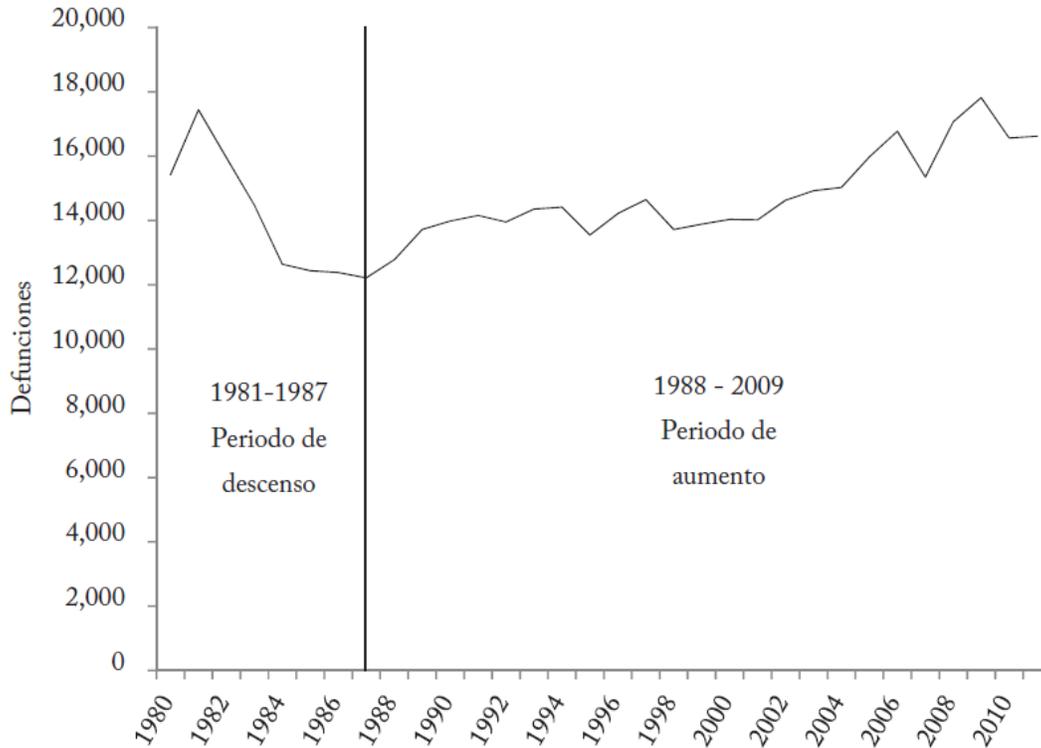
En México cada año fallecen en promedio 16 700 personas⁹ debido a accidentes viales y para el año 2013, el país se encontraba en la posición 98 de 198 países con una tasa de mortalidad de 14.4 por cada cien mil habitantes.

La Gráfica 3.1.1, contenida en el Tercer Informe sobre la Situación de la Seguridad Vial, muestra la tendencia en cuanto a las defunciones debido a siniestros de tránsito en México en dos períodos. Un primer período destaca por una disminución de fallecimientos hasta cerca de 12 mil personas en el año 1987. A pesar de los intentos en el país por reducir la cantidad de víctimas, el segundo período nos muestra que el número de muertos ha ascendido hasta más de 16 mil personas por año.¹⁰

⁸ http://www.who.int/gho/road_safety/mortality/en/

⁹ http://www.who.int/violence_injury_prevention/road_traffic/countrywork/mex/en/

¹⁰ Tercer Informe sobre la Situación de la Seguridad Vial, CONAPRA. Pág. 1



Gráfica 3.1.1 Tendencia de la mortalidad por siniestros de tránsito en México, 1980-2011, CONAPRA

Fuente: Tercer Informe sobre la Situación de la Seguridad Vial, CONAPRA. Pág. 9

En el año 2011, de las 16 615 personas que perecieron en accidentes de tránsito, 5 991 fueron usuarios vulnerables, es decir, peatones, ciclistas y motociclistas. La Tabla 3.1.1 muestra cuántos usuarios vulnerables resultaron accidentados, heridos o muertos entre 2006 y 2011.

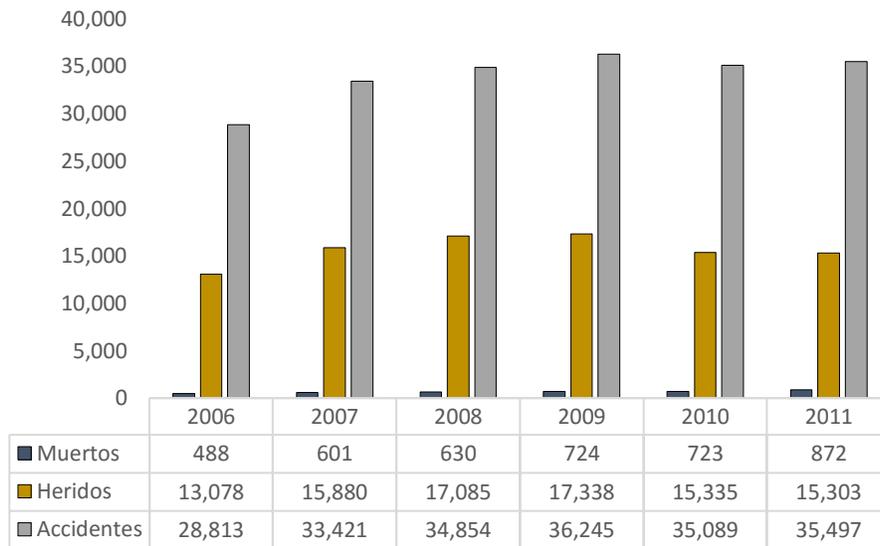
Tabla 3.1.1 Usuarios vulnerables accidentados, 2006-2011¹¹.

Usuarios		2006	2007	2008	2009	2010	2011
Muertos	Peatones	5,2220	4,819	4,934	5,147	4,786	4,868
	Ciclistas	283	234	256	306	261	251
	Motociclistas	488	601	630	724	723	872
Heridos	Peatones	19,713	21,030	19,037	18,490	18,031	16,125
	Ciclistas	6,512	5,931	5,757	5,245	4,476	4,531
	Motociclistas	13,078	15,880	17,085	17,338	15,335	15,303
Accidentados	Peatones	20,991	23,002	20,025	20,546	20,260	18,195
	Ciclistas	7,943	7,290	6,941	8,363	7,583	7,453
	Motociclistas	28,813	33,421	34,854	36,245	35,089	35,497

¹¹ Tercer Informe sobre la Situación de la Seguridad Vial, CONAPRA.

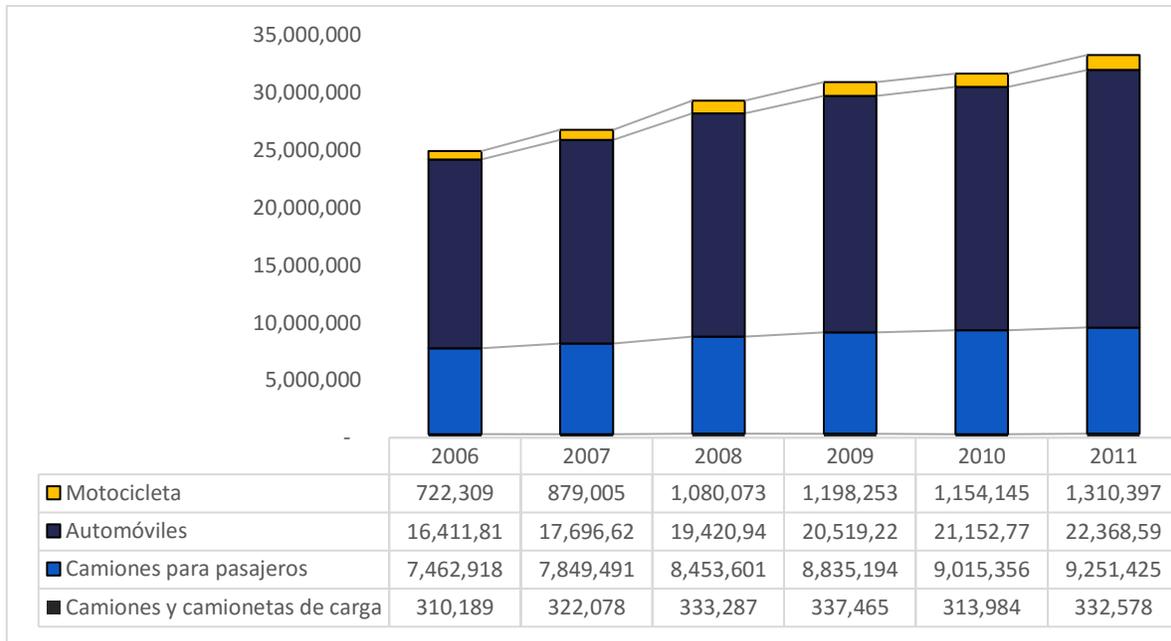
DESCRIPCIÓN Y DESARROLLO

Haciendo un análisis a simple vista se puede determinar que el número de accidentes de tránsito en el caso de los motociclistas aumentó considerablemente en estos seis años y como consecuencia, el número de muertos en estos años casi se duplicó.



Gráfica 3.1.2 Representación gráfica de la Tabla 3.1.1.

En México, el aumento del uso de motocicletas está directamente relacionado con el menor costo de adquisición y consumo de combustible, además de ser una solución al problema del congestionamiento y del estacionamiento, entre otras razones. Este incremento queda evidenciado en la Gráfica 3.1.3.



Gráfica 3.1.3 Aumento del número de vehículos en México, 2006-2011.

Como consecuencia, también aumentaron los accidentes de motocicletas en el país siendo la principal razón de los fallecimientos la poca protección que ofrecen estos vehículos en cuanto a las estructuras y sistemas, mismos que sí ofrecen los coches, camiones, etc.

Un parámetro que de no cumplirse constituye un factor de riesgo es la distancia de frenado. Esta es la distancia que le lleva a un vehículo para pasar de cierta velocidad a un alto total. Esta medida es importante en la rama relacionada al tránsito vehicular pues ayuda a determinar el espacio que deben de guardar los conductores entre sus vehículos para una cierta velocidad previamente establecida. Para nuestro proyecto, estas medidas son útiles para determinar si nuestro dispositivo sería capaz de detectar el patrón de un frenado común y corriente al de un accidente.

Utilizando las ecuaciones del movimiento, podemos llegar a $t = \frac{2d}{(v+v_0)}$ ¹², la cual nos ayuda a determinar el tiempo de frenado de un vehículo dadas sus distancias de frenado. Nuestro

¹² donde: t = tiempo, d = distancia, v = velocidad final, v_0 = velocidad inicial.

DESCRIPCIÓN Y DESARROLLO

GPS actualiza su posición cada segundo, por lo que para detectar si ocurrió un accidente midiendo las velocidades, el tiempo de frenado tendría que ser mayor a esto.

Conociendo que la distancia de frenado promedio de las motocicletas ronda los 140 pies (45.7 m) a 60 millas por hora (96.5 km/h o 26.8 m/s), se tiene que el tiempo de frenado promedio es de 3.41 segundos.

Como puede observarse, esto da únicamente 3 actualizaciones de velocidad antes de llegar al alto total, por lo que el querer determinar el patrón de un frenado normal y el patrón de un accidente puede ser muy arriesgado. Por esta razón, la elección de sensores sigue siendo sumamente importante.

Otros factores de riesgo al conducir una motocicleta son:

- Edad

Según estudios realizados por la OMS para su Tercer Informe, en el año 2011 el 67.6 % de los muertos en accidentes de motocicletas se concentraba entre las edades de 15 y 34 años, evidenciando esto cómo las conductas de riesgo de los jóvenes constituyen el principal factor de riesgo.

- Alcohol

A pesar de que la cifra de personas fallecidas con aliento etílico oscila entre un 10 y un 15 %, esta cuestión se ubica como el segundo factor de riesgo.

- Temporada

- Día de la semana

- Hora
-

Además, el documento “Los accidentes de motocicleta en México”, redactado y difundido por la CENAPRA, menciona que “(...) uno de los principales factores de riesgo conocidos es la velocidad, sin embargo, los registros oficiales en México no consideran a este factor por lo tanto no es medido”.

Tras una revisión de los reglamentos o leyes de tránsito para cada uno de los estados de la República, se observa que en la mayoría de estos se consideran las mismas medidas de control, tales como:

- Funcionamiento de luces y frenos.
- Uso de casco para el conductor y acompañante.
- Solo algunos estados consideran el examen teórico y práctico de manejo.
- Algunos estados como Durango, Coahuila, entre otros, permiten la conducción de motocicletas a menores de 14 años.
- Regulación de la circulación en los carriles de la extrema derecha. El Distrito Federal prohíbe la circulación en vías primarias.¹³

Del Reglamento de Tránsito del Distrito Federal:

III. Adicionalmente, los motociclistas deben:

- a) Circular todo tiempo con las luces traseras y delanteras encendidas;*
- b) Llevar a bordo sólo la cantidad de personas para las que exista plaza disponible;*
- c) Usar aditamentos luminosos o bandas reflejantes en horario nocturno;*
- d) Utilizar casco protector diseñado específicamente para motociclistas y asegurarse que los acompañantes también lo usen; éste debe estar correctamente colocado en la cabeza y abrochado; y*

¹³ Reglamento de Tránsito del Distrito Federal y Aviso; Gaceta Oficial del Distrito Federal; Número 156 Bis

DESCRIPCIÓN Y DESARROLLO

e) Preferentemente portar visores, chamarra o peto para protección con aditamentos rígidos para cobertura de hombros, codos y torso específicos para motociclista, guantes y botas, todos de diseño específico para conducción de este tipo de vehículo.

Haciendo referencia al punto *d*, en México el uso del casco de seguridad para motociclistas no es de uso general, de hecho, según cifras de la Federación Mexicana de Motociclistas, cerca del 90 % de los conductores no lo utilizan, incluso cuando el uso del mismo es un factor decisivo en casos de accidentes con un 40 % más de probabilidades de sobrevivir en comparación con quien no lo lleva.

En una entrevista concedida por Miguel Valdés, expresidente de la Federación Mexicana de Motociclistas, y publicada por El Economista en enero de 2013, este destaca que “(...) *el casco [para motociclista] tiene la función primordial de proteger el cerebro y la cara. Debe reunir una serie de requisitos que están plasmados en normas que expiden las autoridades en Estados Unidos o Europa, porque en México no existen*”.

Como se destacó anteriormente, en el país sí existe una regulación en cuanto al uso del casco de seguridad por parte de las autoridades pertinentes, sin embargo, no existe una Norma Oficial que regule las características que debe tener un casco para la protección del usuario.

3.2 Análisis del problema

3.2.1 Propuesta de solución al problema

El método para llegar al diseño propuesto siguió los siguientes pasos:

- Análisis de los accidentes en motocicleta
- Análisis del funcionamiento ideal del dispositivo y acotamiento de su alcance
- Análisis de las herramientas con las que se podrían disponer
- Desarrollo del prototipo
- Pruebas del prototipo

Este método brinda la suficiente visión sobre el problema y la forma de solucionarlo, así como un marco establecido de pasos a seguir para llegar a la solución más óptima.

Análisis de los accidentes en motocicleta

Los accidentes de motocicleta, comparados con los accidentes automovilísticos, suelen ser proporcionalmente más fatales. Se estima que en 2013, el número de muertes provocadas por accidentes en motocicletas era 27 veces mayor a las provocadas por accidentes en automóvil. Además, el porcentaje de accidentes de vehículo único, son cerca del 42 % de los casos en Estados Unidos.

Aproximadamente el 25 % de los accidentes fatales de motocicletas se concentran en las horas de 9 pm a 6 am, y el 46 % de estos accidentes ocurren en zonas rurales.¹⁴

Dadas las condiciones mencionadas, y haciendo un análisis de estos datos, podríamos inferir que la persona accidentada probablemente no tuvo la oportunidad de pedir auxilio a las autoridades competentes en un tiempo razonable, si es que se pidió. Aunque no se tienen

¹⁴ <http://www.iihs.org/iihs/topics/t/motorecycles/fatalityfacts/motorecycles>

DESCRIPCIÓN Y DESARROLLO

estadísticas del número de vidas que se habrían salvado en caso de recibir atención médica inmediata, si se aplica el concepto de “hora de oro”, indica que la hora posterior a un accidente que pueda causar la muerte es crítica para las acciones que puedan prevenir la muerte, podemos entender que si alguien sufre un accidente y no se da aviso a los servicios de emergencia, las acciones que se pudieran llevar a cabo para salvar la vida de los pasajeros se ve reducida.

Análisis del funcionamiento ideal del dispositivo

El diseño del dispositivo debe cumplir con las siguientes características:

- Operar sin intervención humana
- Comunicarse con los servicios de emergencia
- Proporcionar información del lugar del accidente a estos servicios
- Proporcionar información médica relevante del posible accidentado
- Continuar operando a pesar del impacto del accidente

Debido al enfoque de este trabajo, nos enfocaremos en los primeros cuatro puntos, dejando de lado aquellos que otras disciplinas podrían resolver de mejor manera, como el diseño de un empaque para resistir los impactos que pudiese sufrir el dispositivo o la definición de la información médica relevante al momento de sufrir un accidente.

Sin embargo, para todos aquellos puntos los cuales el alcance de este trabajo no cubra, se procurará establecer una guía de diseño previa con el fin de establecer los parámetros básicos en los que debiera operar.

Autonomía

La premisa básica es que este aparato sea un dispositivo autónomo, es decir que no requiera de la operación o intervención humana para lograr su objetivo. Por lo tanto, se define que debe cumplir con lo siguiente:

- Operar de manera pasiva
- Diagnosticar el momento en el que ocurra un accidente vial
- Contar con un método que evite los falsos positivos
- Recopilar información y procesarla para poder comunicarla
- Tomar decisiones que permitan una mejora de la situación

Para lograr todo lo anterior, se decidió que la mejor manera de lograrlo era con un dispositivo electrónico programable, que contara con la capacidad de ejecutar algoritmos de toma de decisión simple, así como tener la capacidad de comunicarse con diversos subsistemas que proporcionen la suficiente información para determinar si ocurrió un percance y los datos necesarios mínimos para los servicios de emergencia.

Debido a las características de operación especificadas anteriormente, el dispositivo debe estar presente en el vehículo en el momento de un accidente. Para ello, el sistema deberá ser capaz de operar con fuente de energía propia y poder permanecer operando por lo menos un día completo.

Tomando en cuenta todos estos factores, se determinó que el dispositivo central debería de ser un microcontrolador, ya que este tipo de aparato electrónico cumple con todo lo necesario para tener una base de trabajo funcional.

DESCRIPCIÓN Y DESARROLLO

Comunicación

La operación básica del dispositivo en cuestión es detectar un accidente, recopilar información y comunicar de manera efectiva la información recopilada. Por lo tanto, la comunicación, y por tanto envío, de los datos recopilados es de máxima importancia.

Los factores que se deben de considerar son los siguientes:

- La comunicación que establezca el dispositivo debe ser instantánea o cuasi-instantánea, es decir, debe permitir que el intercambio de información se desarrolle en un tiempo no mayor a 1 minuto después de ocurrido el percance.
- La comunicación debe de establecerse en prácticamente cualquier región.
- La comunicación, por naturaleza misma del objetivo, debe ser móvil.
- Debe usarse un sistema de comunicación común y confiable, evitando así un mayor costo del dispositivo final.

Por los factores anteriormente mencionados, la comunicación celular se propuso como la mejor solución ante estos desafíos. La telefonía celular es una tecnología que actualmente está lo suficientemente madura como para poder confiar en ella, y además cuenta con la suficiente cobertura nacional como para cumplir con nuestros propósitos.

La cobertura de los servicios celulares es de suma importancia para este tipo de dispositivos, ya que no se puede conocer a ciencia cierta el lugar en el que podría ocurrir el accidente. Aunque las tecnologías más recientes están ampliando su cobertura, la tecnología 2G es la que tiene un mayor porcentaje de área cubierta. Por lo tanto, se escogió esta tecnología en su versión GSM/GPRS, la cual es compatible con la red de Radiomóvil Dipsa S.A. de C.V. (Telcel) así como con otras compañías.

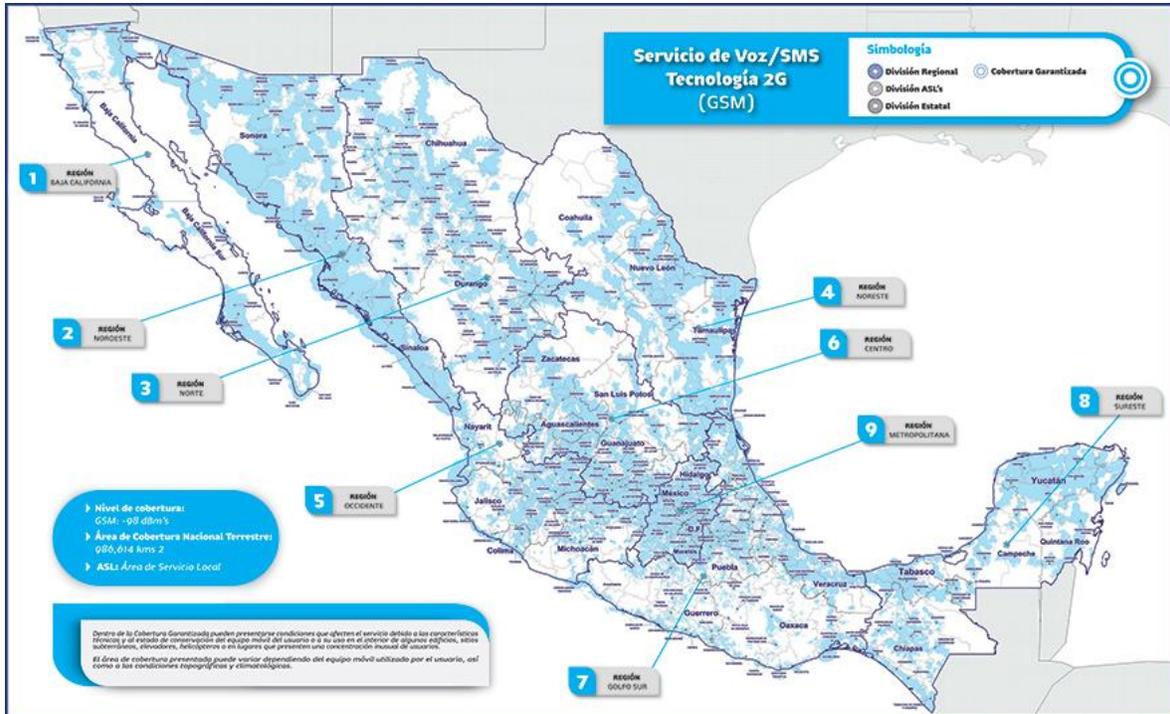


Fig. 3.2.1 Cobertura de la red 2G en México. Fuente: Telcel.

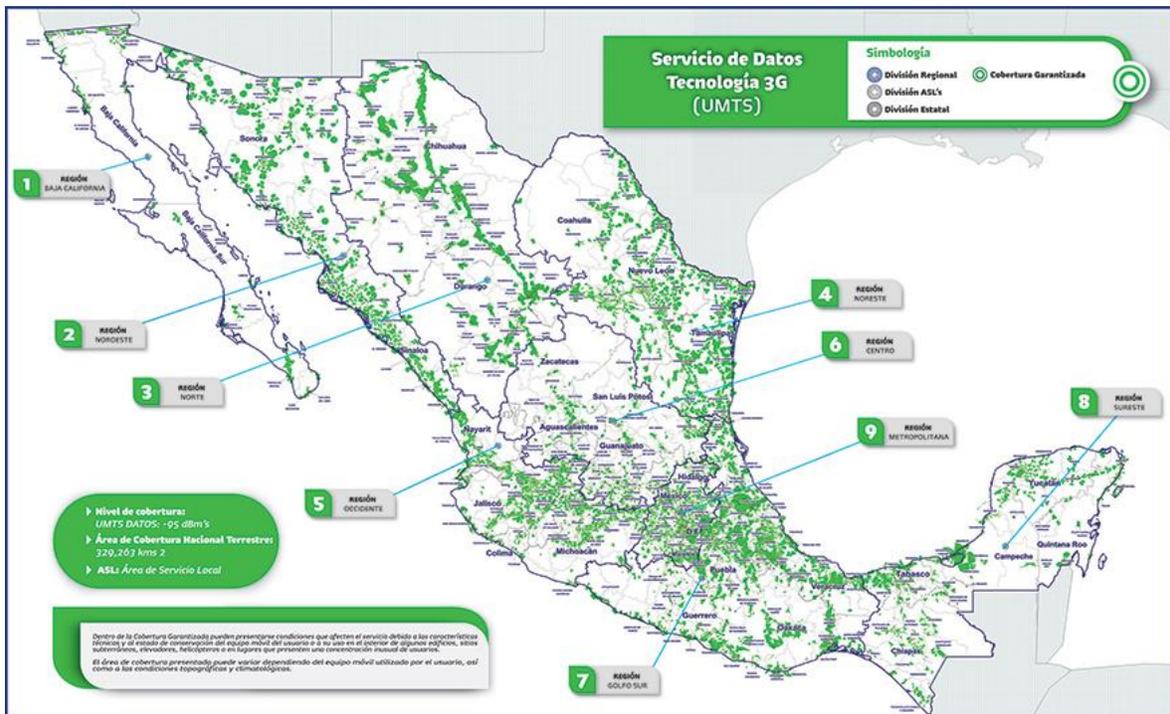


Fig. 3.2.2 Cobertura de la red 3G en México. Fuente: Telcel.

DESCRIPCIÓN Y DESARROLLO

Información del accidente

El dato más importante para lograr el objetivo planteado con el diseño de este dispositivo es la ubicación del accidente. Sin este dato clave, los servicios de emergencia son incapaces de brindar asistencia médica en caso de un percance, por lo que es crítico que se pueda obtener, procesar y transmitir la posición geográfica del dispositivo en cualquier momento.

Para lograr esto, se recurrió a un GNSS (*Global Navigation Satellite System*). Este sistema cumple con todas las características necesarias para conocer la ubicación del percance, además de funcionar de manera autónoma al sistema principal y de manera constante, es decir, que no importa que otros subsistemas de menor nivel crítico fallen, el GNSS podría seguir funcionando sin importar en qué momento se requiera de su uso.

Información médica relevante

A pesar de no ser parte esencial del funcionamiento del dispositivo, consideramos importante la posibilidad de poder almacenar datos relevantes, tales como tipo de sangre o alergias a medicinas, para una mejor atención al usuario.

Debido al restringido espacio con el que se cuenta para almacenar y transmitir la información, se deben elegir solamente dos datos que sean de vital importancia y que puedan ayudar a una mejor atención del paciente.

Continuidad en la operación

Como parte importante del diseño del dispositivo, se encuentra la capacidad de continuar operando de la manera establecida a pesar de no contar con una fuente de energía constante.

Por ello, el dispositivo debe ser capaz de operar con baterías portátiles que ofrezcan una autonomía suficiente como para que el dispositivo cumpla con su objetivo principal. Como valor añadido, las baterías deberían de ser recargables.

Si fuera posible, una fuente de luz solar ofrecería la ventaja de liberar al usuario de la responsabilidad de mantener su dispositivo con la suficiente batería para funcionar, lo cual evitaría que en caso de olvido o descuido, el funcionamiento del dispositivo no se vea comprometido.

Resumen de características básicas

Como se explicó anteriormente, el dispositivo debe de cumplir con las siguientes características básicas:

1. Estar compuesto por un microcontrolador, el cual fungirá como pilar del diseño del dispositivo. El microcontrolador en cuestión debe ser capaz de conectarse con sistemas periféricos, procesar información y tomar decisiones en tiempo real.
2. Contar con un módulo de comunicación celular, con capacidad de conectarse a las redes 2G.
3. Contar con un módulo de localización GNSS que sea compatible con el sistema GPS.
4. Capacidad suficiente de almacenamiento para guardar los datos relevantes del usuario.
5. Poder operar con baterías.

3.2.2 Decisiones de diseño

Para propósitos de este proyecto, y para poder cumplir con las características anteriormente mencionadas, se tomaron múltiples decisiones de diseño, especialmente en la elección de los módulos con los que se trabajaría.

Cabe mencionar que para elegir entre distintos modelos de aparatos con los que se trabajaría se tomaron en cuenta varios parámetros listados a continuación:

1. Cumplimiento básico de las funciones necesarias

DESCRIPCIÓN Y DESARROLLO

2. Disponibilidad del dispositivo en México
3. Compatibilidad con el resto del sistema
4. Facilidad de trabajo con otros dispositivos
5. Comparativo contra otros aparatos similares
6. Costo

Elección de microcontrolador

La elección del microcontrolador es probablemente la más sencilla, pero al mismo tiempo el proceso más extenso. Debido a que esta época está caracterizada por una amplia oferta de microcontroladores en todo el mercado, y dado que un gran número de ellos empiezan a trabajar de manera muy similar, muchas veces pueden ser intercambiados sin problemas.

Desde hace algunos pocos años, surgió un movimiento que acercó la programación de microcontroladores a toda la comunidad entusiasta, el Arduino. La principal característica de esta plataforma es que, a diferencia de lo que existía en el mercado, las personas no debían tener un gran conocimiento de electrónica para hacerlo funcionar y poder realizar sus proyectos.

Debido al éxito que tuvo Arduino, muchas empresas comenzaron a ofrecer productos similares. Entre ellas se encuentran Intel, Raspberry Pi y Texas Instruments. Cada una de ellas ofrece una oferta de valor diferente, pero la premisa sigue siendo la misma: hacer de la electrónica algo que no sea únicamente para especialistas.

Se hizo una comparativa entre 2 de las plataformas más comunes y que cumplían con el propósito del proyecto: Arduino Uno y Texas Instruments MSP430 Launchpad.

El Arduino Uno, como ya se mencionó previamente, es una tarjeta de desarrollo sumamente popular, con una gran comunidad de soporte y basado en el *software* libre. Su

microcontrolador es un ATMEL MEGA328p, y cuenta con un gran número de librerías con las que se pueden desarrollar proyectos de manera muy sencilla.

El MSP430 Launchpad es una alternativa provista por Texas Instruments, la cual busca hacer la competencia a Arduino y servir como puerta de entrada al resto de sus productos. La serie Launchpad toma mucho de lo desarrollado por Arduino y lo adapta a su plataforma. Ofrece tarjetas de desarrollo basadas en distintos chips, siendo el más prominente el MSP430 en diferentes versiones.

Para nuestro caso, se compararán las características del Arduino Uno R3 y del MSP430F5529 Launchpad.

A continuación haremos una comparación de ambas plataformas, y se explicará la decisión que se tomó basándonos en las ventajas y desventajas de cada una de ellas.

Cumplimiento básico de las funciones necesarias

Este apartado más allá de otorgar un calificativo de mejor o peor a las plataformas es el primer paso para descalificar a alguna dado el caso de que no cumplieren con las funciones necesarias. Para recordar, las funciones necesarias esperadas del microcontrolador son:

- Capacidad de tomar decisiones
- Capacidad de funcionar con batería
- Capacidad de conectar dispositivos externos

La capacidad de tomar decisiones se considera como aquella en la que el dispositivo es capaz de ejecutar algoritmos. La capacidad de funcionar con batería muestra que el dispositivo es capaz de operar con fuentes de corriente directa que pueden ser independientes a la red de energía del vehículo, por ejemplo. La capacidad de conectar dispositivos externos es aquella en la que el dispositivo está habilitado para comunicarse con dispositivos externos.

DESCRIPCIÓN Y DESARROLLO

El Arduino Uno R3, como se mencionó antes, funciona con un microcontrolador Atmel ATMEGA328p, el cual es capaz de ejecutar comandos que le sean programados. Además, el consumo de energía es lo suficientemente bajo como para poder operar con baterías, y la tarjeta está preparada para conectarse con una fuente de energía externa, ya que cuenta con una entrada para este propósito. La tarjeta de desarrollo cuenta con diversos protocolos de comunicación serial, y además, gracias a los pines de GPIO (Entrada/Salida de Propósito General – *General Purpose In/Out*) y a una librería de Serial de Software, la tarjeta es capaz de emular conexiones tipo serial en un mayor número de pines.

De igual manera, el Launchpad cuenta con un microcontrolador, este de la marca Texas Instruments, que también es capaz de ejecutar comandos que le sean programados. A pesar de no tener una entrada específica como fuente de energía, su conexión microUSB le permite utilizar fuentes externas, y dado su bajo consumo de energía, es posible alimentarlo con una batería. Al igual que el Arduino, el Launchpad cuenta con varios pines habilitados con protocolos de comunicación serial, y con su propia librería de Serial de Software, con el cual también es capaz de emular conexiones de este tipo.

Como vemos, ambas plataformas cumplen con los requisitos básicos de funcionamiento.

Disponibilidad de dispositivos en México

Para poder llevar a cabo el proyecto, es necesario que los materiales necesarios estén disponibles en donde se van a utilizar. Por lo tanto, es importante conocer si ambos dispositivos son accesibles para el mercado mexicano.

La plataforma de Arduino Uno, dada su popularidad, se encuentra disponible en muchas tiendas de electrónica, así como en venta en tiendas en línea. El precio oscila los 300 MXN, por lo cual no sólo tiene disponibilidad, sino también accesibilidad.

El *launchpad* de Texas Instruments no es tan popular, por lo que encontrarlo en tiendas físicas no es muy común, pero la compañía lo pone a disposición en su tienda en línea, y es posible recibir el producto en el país. Dependiendo del modelo, los precios van desde los 9.99 USD hasta los 29.99 USD, por lo que el precio es mucho más competitivo que el de Arduino.

Compatibilidad con el resto del sistema

A pesar de todas estas similitudes, la principal razón por la que nos inclinamos por el sistema Arduino tiene que ver con un tema de compatibilidad con los módulos GSM y GPS necesarios para llevar a cabo el proyecto. El problema del sistema Launchpad es que su comunicación serial se realiza a 3.3 V, y no a 5 V como es estándar para los módulos de Arduino, por lo que el correcto funcionamiento de los módulos finalmente seleccionados se vio comprometido.

Elección del GPS, SIM900, Telcel

GPS

En el mercado existen muchas soluciones de módulos de experimentación que permiten recibir y procesar datos del GNSS. La mayoría de ellas, sin embargo, son difíciles de encontrar en existencia, o simplemente no están adaptadas para una conexión sencilla a los distintos módulos de desarrollo.

En las primeras etapas de nuestra búsqueda de un módulo, nos enfocamos en buscar soluciones desarrolladas por Texas Instruments, debido a que suponíamos que sería una integración mucho más sencilla.

Para los modelos Launchpad y sus respectivos módulos de conexión, no hay ninguno que cumpla con la función de proveer información de geolocalización. De esa compañía, el módulo que surge como primera opción es el módulo CC4000-TC6000GN, el cual está

DESCRIPCIÓN Y DESARROLLO

desarrollado para funcionar para todos aquellos dispositivos que requieran una ubicación precisa. Este módulo se conecta a través del protocolo serial UART, tiene una sensibilidad de -162 dBm, envía la información de acuerdo a lo establecido por el protocolo NMEA, y ofrece datos de latitud, longitud, tiempo y velocidad. La única desventaja es que este módulo se diseña para montarse sobre la tarjeta *MSP430F5529 USB Experimenter's Board*.

A primera vista, este módulo parecía la mejor opción, así que se decidió comprarlo. El problema de no estar diseñado para los modelos Launchpad parecía poco importante, debido a que siempre se puede conectar vía cables *jumper*, y se esperaba el mismo desempeño.

Lamentablemente, al recibir este módulo, se buscó hacer la conexión vía *jumpers*, misma que no se logró debido a que las conexiones físicas del módulo eran de un calibre completamente distinto a los necesarios para establecer la conexión con el Launchpad. Además, realizar las conexiones con los calibres necesarios no permitió un funcionamiento correcto de los dispositivos, por lo que se tuvo que desechar la idea de utilizar el módulo para este proyecto.

Después de esta etapa, se analizaron distintos proveedores que manejaran dispositivos con una capacidad similar, pero que pudieran conectarse sin tanta dificultad a los módulos. Como siguiente opción surgió el módulo *SparkFun GPS Shield Kit*, diseñado originalmente para funcionar con Arduino. Este módulo incluye un chip EM-506 de la marca GlobalSat, y funciona con el protocolo serial UART, tiene una sensibilidad de -163 dBm, y cumple con lo establecido en el protocolo NMEA. El tiempo de fijación es menor a 35 segundos, lo cual es de gran utilidad para nuestro proyecto.

Este módulo, al venir sin ensamblarse, permite una mayor libertad al momento de buscar la conexión adecuada con el módulo. Además, a diferencia de aquel de la marca Texas Instruments, nos permitió una conexión mucho más amigable con ambas plataformas, tanto

la plataforma Arduino (conexión directa) como la plataforma Launchpad (a través de cables *jumpers*)

A pesar de que el módulo GPS de *Sparkfun* cumplía los requisitos del proyecto, se decidió buscar un modelo que tuviera la posibilidad de montarse sin la necesidad de *jumpers* en un Arduino, y que no interfiriese con el módulo GSM. Además, se buscaba un dispositivo más robusto ya que en pruebas preliminares se presentaron algunas fallas.

En el mercado se encontró el módulo *Tinysine GPS Logger Shield V2*, el cual nos ofrece una sensibilidad de -165 dBm, con actualizaciones de 10 Hz y 66 canales. Además ofrece la posibilidad de grabar los datos dentro de una memoria SD, lo cual no se explotó en este proyecto pero que permitiría tener funcionalidades adicionales sin necesidad de otro módulo.

Librería TinyGPS++

Gracias a la conexión tipo UART de este módulo, extraer las oraciones NMEA es sencillo. Sin embargo, las oraciones NMEA que contienen la información que necesitamos deben pasar por un proceso para extraer los datos de la oración. Este proceso, conocido coloquialmente como *sentence parsing*, requiere que el microprocesador corra algoritmos para poder extraer los datos.

Debido a que este es un proceso común, la comunidad de Arduino ha generado una librería escrita para esta plataforma que hace el *sentence parsing* por nosotros. Esta librería es conocida como **TinyGPS++**, y es gratuita y de código libre.

Como se explicó anteriormente, esta librería se encarga de procesar los caracteres recibidos a través del serial UART, los procesa y nos devuelve la información extraída. Además, se encarga de convertir las unidades, por ejemplo, convierte la velocidad a kilómetros por hora, o el tiempo en horas, minutos y segundos.

DESCRIPCIÓN Y DESARROLLO

Esta librería, a pesar de estar diseñada y codificada para Arduino, después de varios intentos, pudo ser anexada sin requerir ninguna modificación de nuestra parte al sistema de *Energia*, la plataforma en la que se escribe el código para los sistemas Launchpad, y además se logró que el resultado fuera correcto.

Además de esta librería, es beneficioso añadir la librería *SoftwareSerial*, incluida en ambas plataformas por defecto. Esta librería nos permite conectar el módulo GPS a puertos que no necesariamente están diseñados para funcionar con el protocolo UART y poder establecer una comunicación. Es especialmente útil en la etapa de pruebas, ya que permite comunicarnos vía PC con el dispositivo en cuestión sin que este deje de comunicarse con el módulo.

Módulo de comunicación GSM

El módulo GSM no es algo tan común como uno esperaría en esta época. Al inicio de nuestra investigación queríamos hacer uso de las tecnologías más nuevas disponibles en el mercado, como lo son los sistemas 4G y la conectividad celular de mayores anchos de banda. Lamentablemente, este tipo de módulos no se encuentran disponibles como módulos extras a tarjetas de experimentación. Compañías tales como Qualcomm y Samsung están trabajando en crear tarjetas con este tipo de conectividad, pero se encuentran en desarrollo (Samsung y la línea *Artik*) o a precios sumamente elevados (Qualcomm y Digi y el kit *Internet of Everything Kit by Etherios*).

La búsqueda del módulo ideal comenzó en la misma compañía que los módulos anteriores, Texas Instruments, esta vez sin darnos ningún tipo de resultado útil para nuestro proyecto. Esta compañía cuenta con una línea de chips enfocados en la comunicación GSM y GPS

(TCS2600), pero se centra en fabricantes de celulares, por lo que no cuenta con tarjetas de prueba que sean compatibles con los módulos requeridos.

La segunda opción fue el módulo GSM distribuido por Arduino. Este módulo está basado en el chip M10 de Quectel, el cual permite hacer uso de todas las funcionalidades brindadas por la segunda generación de comunicaciones celulares. Además de proveer una librería nativa en el sistema para hacer uso de este módulo, es posible establecer la comunicación a través de los comandos AT. Este módulo cumple con todas las funcionalidades necesarias para llevar a cabo el proyecto, sin embargo, durante toda la realización del mismo, no hubo unidades disponibles para su venta, por lo que tuvimos que buscar otra opción.

Finalmente, se recurrió a uno de los productos provistos por la compañía Epalsite. En este caso, el módulo fue el *GPRS Shield VI.1(B)*. Este módulo, basado en el chip SIM900 de la compañía SimCom, está diseñado para ser compatible con Arduino, y por lo tanto, se dedujo que funcionaría también con los dispositivos Launchpad. Cuenta con la capacidad de comunicarse con el módulo vía UART y vía los comandos AT, y se puede adaptar la librería GSM del Arduino para que funcione con este módulo. Además, cuenta con una antena externa omnidireccional.

Al realizar las pruebas con este módulo nos encontramos con un gran número de dificultades, mucho mayores que los encontrados en el módulo GPS. El primero de ellos fue encontrar la manera en la que los comandos AT pudiesen funcionar de manera correcta, y de manera subsecuente que el módulo funcionase cuando se requiriese.

Al no encontrar una mejor alternativa, decidimos trabajar con este módulo.

Elección de proveedor de Comunicación celular

Como parte importante del funcionamiento del proyecto está la elección del proveedor de comunicación celular. Para esta sección, y dado que es relativamente sencillo cambiar de un momento a otro de proveedor sin afectar de manera grave el funcionamiento del dispositivo, la elección estuvo basada principalmente en dos razones. La primera es que se pudieran establecer comunicaciones fácilmente con este proveedor y la segunda es el costo en el que incurriríamos al realizar las pruebas.

El proveedor Radiomóvil Dipsa, mejor conocido como Telcel, fue nuestra elección ya que cumplía las dos necesidades requeridas.

El hecho de que los chips de esta compañía funcionasen correctamente se debe en buena medida a que logramos encontrar una tarjeta SIM de tamaño completo, de la versión 5.0, como requiere el módulo GSM. Inicialmente, se realizaron las pruebas con una tarjeta tipo microSIM v6.1 y un adaptador a tamaño completo, pero el módulo SIM900 no lograba reconocerlo. A continuación, se intentó con una tarjeta nanoSIM v6.2 y un adaptador, obteniendo los mismos resultados. La tarjeta SIM v5.0 de tamaño completo hizo que lográramos establecer una comunicación utilizando la red Telcel.

Para el segundo requerimiento, la compañía ofrece a todos sus usuarios llamadas o envío de mensajes gratuitos a 3 números nacionales, lo cual para los propósitos de este proyecto fue de suma utilidad, pudiendo realizar numerosas pruebas de envío de mensajes y llamadas sin incurrir en elevados costos.

3.3 Implementación

3.3.1 Conexiones y descripción por etapas

Para iniciar los procesos de desarrollo necesitamos asegurarnos de contar con todos los elementos. Para ello existen dos rubros importantes previos a las pruebas de funcionamiento: la parte de *hardware* y la parte de *software*.

Configuración de *hardware*.

Circuito impreso

La necesidad de crear un circuito impreso donde cada *switch* o *push button* funcione como sensor se debe primeramente a las facilidades y ventajas que ofrece con respecto a cables de mayor longitud conectados a un *protoboard*, un segundo aspecto es la complejidad y costo de sensores de impacto.

Tomando esto en cuenta, diseñamos el siguiente circuito con la ayuda del programa *Livewire*.

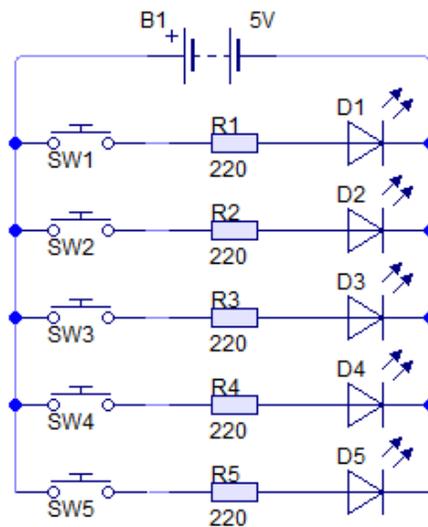


Fig. 3.3.1 Diseño del circuito impreso I

DESCRIPCIÓN Y DESARROLLO

Este circuito se compone por cinco interruptores de los cuales los cuatro primeros funcionan como sensores de impacto. El quinto interruptor cumple dos funciones: la primera de ellas es la de activar la opción de cambiar los datos del usuario vía SMS justo después de encender la motocicleta por un período de un minuto; la segunda opción es la de realizar una llamada a un número predeterminado, preferiblemente a servicios de emergencias, en cualquier instante pasados los primeros 60 segundos luego de haber encendido el vehículo.

Los LEDs incorporados al diseño funcionan solo como indicadores de que ha ocurrido un accidente, es decir, se han activado uno o más sensores. Los cinco contactos que aparecen después de cada *push button* serán las salidas al Arduino, a través de los módulos GPS y GSM, quien interpretará el cambio de voltaje como un incidente automovilístico.

La Figura 3.3.2 pertenece al mismo circuito exportado al programa *PCB Wizard*. Esta será utilizada para el proceso de impresión y planchado sobre la placa de cobre.

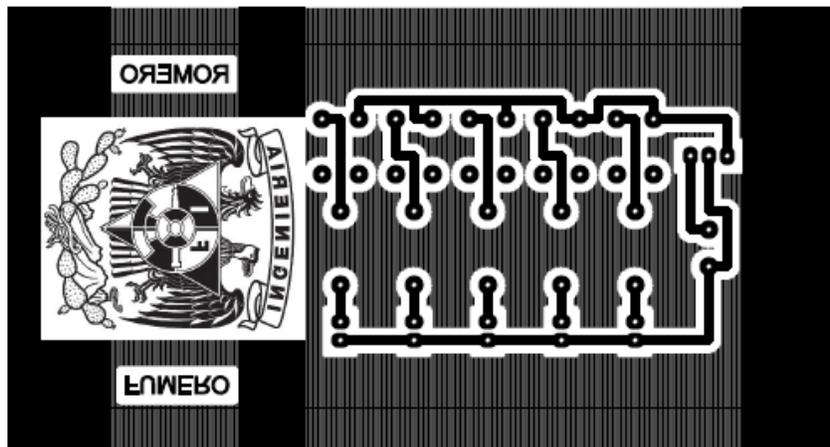


Fig. 3.3.2 Diseño del circuito impreso II

Arduino

Fig. 3.3.3 Vista superior del Arduino Uno

Esta placa de desarrollo cuenta con un puerto de alimentación de 5 Volts de corriente directa y una entrada serial. Esta última será empleada con dos fines: subir el código y correr el programa, y servir de fuente de alimentación tanto al Arduino como al resto de los módulos. Una vez el programa esté en la memoria del microcontrolador, podrá emplearse una fuente externa de alimentación, tal como una batería, para así permitir la movilidad del dispositivo. Gracias a que los fabricantes de los módulos empleados han colocado cada contacto de manera tal que correspondan con los de la placa de desarrollo de Arduino, el tamaño del dispositivo se ve reducido al montar estos uno sobre otro. Este ensamble puede observarse en las figuras que aparecen en los Anexos.

Este último módulo cuenta con su propio procesador así como con una antena para la transmisión y recepción de mensajes y llamadas. De igual manera, cuenta con una ranura para la tarjeta SIM v5.0 o versiones anteriores y con dos botones de encendido y de reinicio. Al conectar el dispositivo a través del puerto serial del Arduino a la computadora será necesario oprimir durante tres segundos aproximadamente el botón de encendido hasta que el LED de color amarillo comience a parpadear. Para determinar si el dispositivo ya está conectado a la red móvil se deberá esperar unos segundos hasta que el mismo LED amarillo parpadee cada tres segundos. El LED de color rojo permanecerá encendido durante todo el proceso ya que este indica que la placa está alimentada.

3.3.2 Configuración de *software*

La configuración necesaria para poder desarrollar el proyecto se basa en el entorno de desarrollo de Arduino, los controladores de Arduino, y las librerías para controlar de una mejor manera los módulos GPS y SIM900. Todos estos componentes se encuentran de manera gratuita en Internet, y en su mayoría son de código abierto.

Entorno de desarrollo Arduino y controladores

Para poder programar los microcontroladores Arduino, es necesario tener instalado el entorno de desarrollo Arduino que podemos encontrar de manera gratuita en el sitio web de la empresa. El número de versión utilizada fue 1.6.1 así como la versión 1.6.7.

Para poder instalar el entorno de desarrollo debemos contar con una computadora con un sistema operativo Windows, Mac OS o Linux. Cada una de estas versiones cuenta con controladores diferentes. El sistema operativo en el que se desarrolla el proyecto fue en Windows 8.1 así como Windows 10, y no se encontraron diferencias en el funcionamiento

DESCRIPCIÓN Y DESARROLLO

durante la migración de sistema operativo. No se intentó probar en sistemas operativos Mac OS ni Linux.

Durante el proceso de instalación del entorno de desarrollo se instalan en paralelo los controladores. Estos controladores son necesarios para poder programar el microcontrolador a través de su placa de desarrollo y permitir la comunicación serial entre la placa y los puertos USB de la computadora. Arduino maneja un paquete de controladores para sus tarjetas Arduino Uno, Mega, etc., por lo que la instalación regular ya incluye los controladores necesarios.

Finalmente, al terminar el proceso de instalación, se nos pide crear una carpeta en la que se guardarán todos los proyectos que se desarrollen para esta plataforma. Además, cada proyecto genera una carpeta dentro de esta misma, por lo que es importante establecer una ruta en la que no genere conflicto con ningún otro programa. Además, se agregará una carpeta con todos los archivos necesarios para que el entorno de desarrollo pueda funcionar, por lo que debemos tener cuidado de no modificar ninguno de estos archivos.

Librerías necesarias

TinyGPS+

La librería para poder controlar de mejor manera el módulo GPS es la TinyGPS+. Esta librería es de código abierto y descarga gratuita. La podemos encontrar en el siguiente link:

<https://github.com/mikalhart/TinyGPSPlus/releases>

La versión que utilizamos es la 0.94b, lanzada el 26 de noviembre de 2013. Esta librería fue creada por Mikal Hart, y en el desarrollo de nuestro proyecto no nos vimos en la necesidad de tener que modificar ninguna de las funciones prediseñadas de la librería.

El proceso de instalación de librerías para Arduino es el siguiente:

1. Descargar la librería en formato .zip o similares.

2. Descomprimir la librería si es necesario.
3. Cambiar de ubicación la librería descargada, colocarla dentro de la subcarpeta *libraries* que encontramos dentro de la carpeta *Arduino* que se creó al momento de instalar el entorno de desarrollo.

A pesar de que el entorno de desarrollo Arduino cuenta con un gestor de librerías, TinyGPS+ no se encuentra dentro de los repositorios que el gestor maneja, por lo que en caso de necesitar cambiar versiones o actualizar, se debe de volver a hacer el proceso anteriormente descrito.

Librería para SIM900

La librería para poder controlar de mejor manera el módulo GPS es la librería *GSM shields*. Esta librería es de código abierto y descarga gratuita. El link para poder obtenerla es el siguiente: <http://www.gsmlib.org/download.html>.

La versión utilizada en el proyecto es la 3.07.1, lanzada en enero de 2014. Esta librería fue creada por un colectivo del mismo nombre, GSM Shields. En el desarrollo de nuestro proyecto no nos vimos en la necesidad de cambiar la librería, pero existe la posibilidad de tener que editar algunas secciones del código para establecer los puertos de comunicación correctos con el Arduino si es que no se utiliza la versión antes mencionada.

DESCRIPCIÓN Y DESARROLLO

Diagrama de bloques

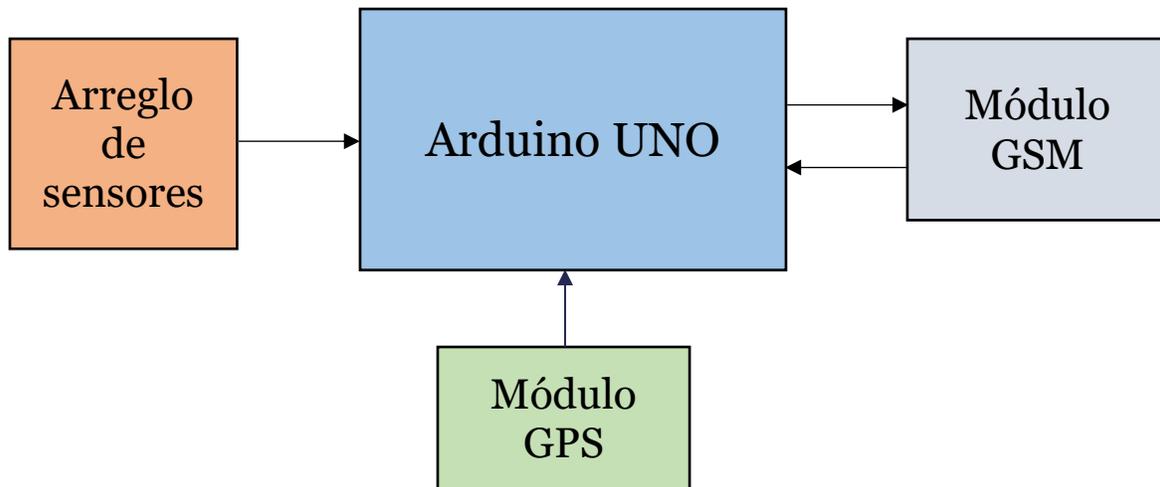
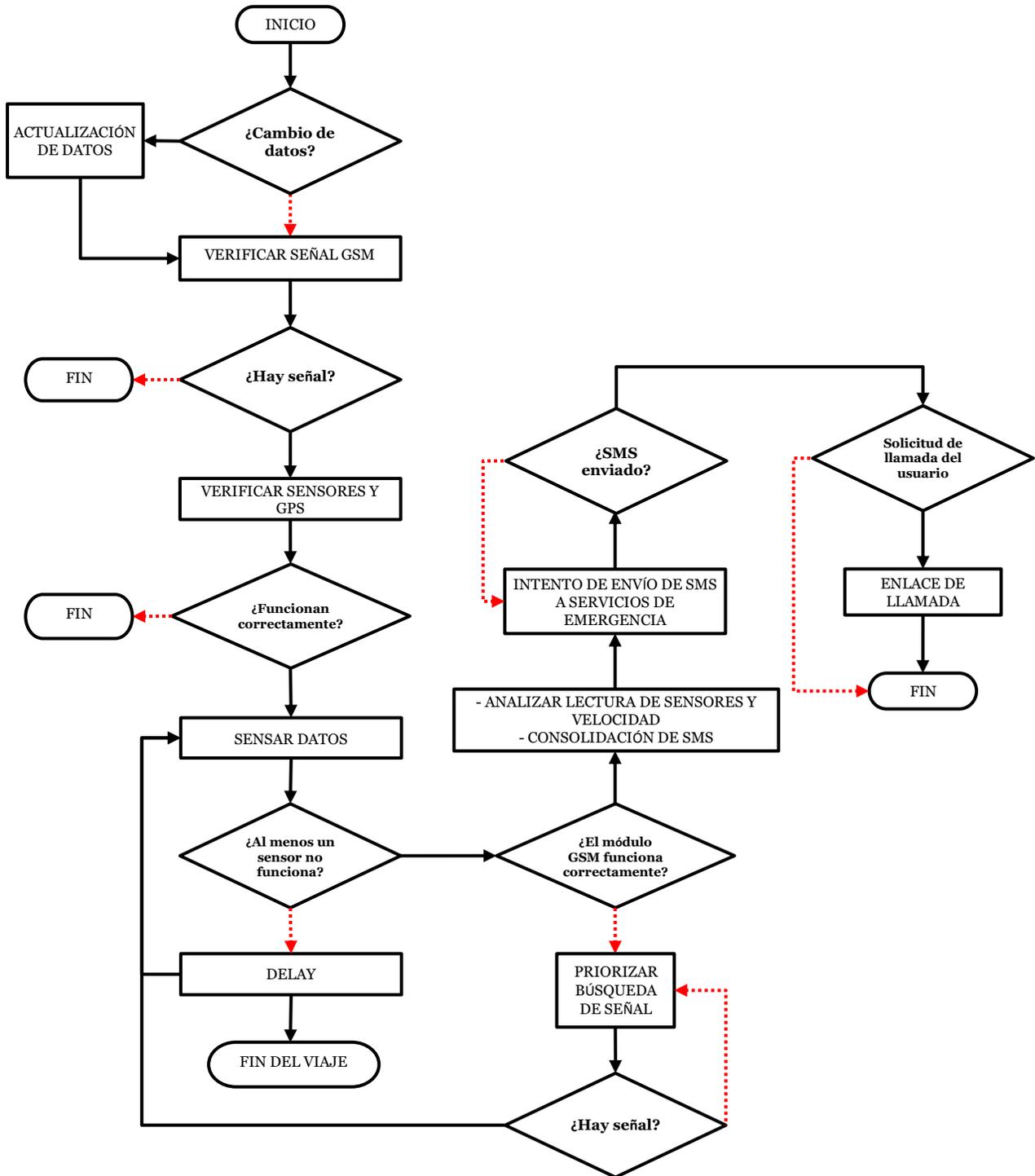
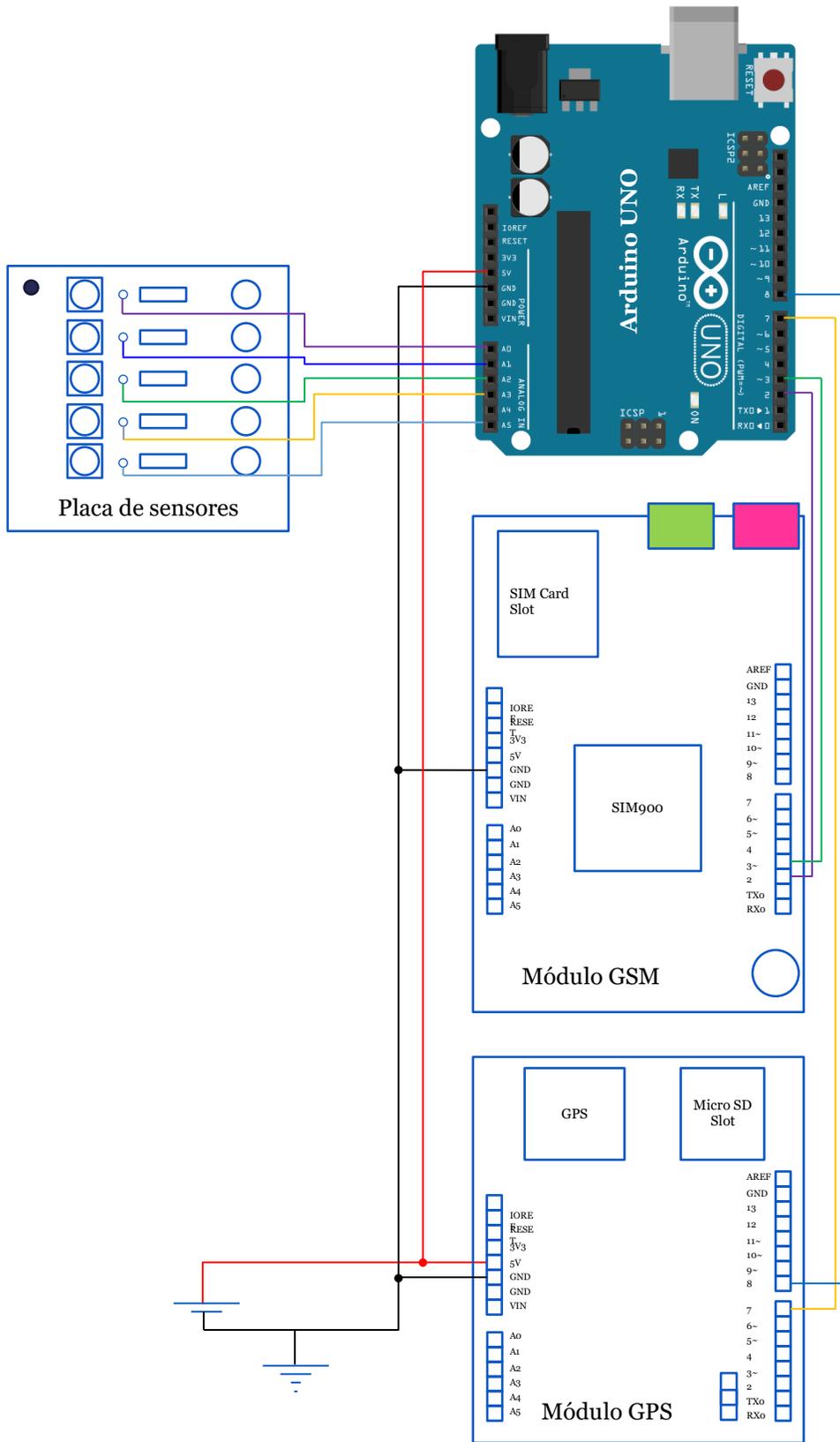


Diagrama de flujo



DESCRIPCIÓN Y DESARROLLO

Diagrama de conexiones



3.4 Pruebas de funcionamiento.

Para confirmar que el funcionamiento de cada uno de los módulos es el adecuado recurrimos a distintos programas de prueba. Además, para los módulos de GSM y GPS recurrimos a una comunicación serial que nos permite enviar directamente comandos a los dispositivos con el fin de establecer una comunicación más directa y comprobar que están realizando su trabajo correctamente.

3.4.1 Pruebas para el Arduino

Las pruebas del Arduino se dividen en dos. La primera es para verificar que el Arduino está funcionando de manera correcta, y la segunda es para comprobar que los puertos que utilizamos estén funcionando también correctamente.

La primera prueba consiste simplemente en cargar el programa *blink*, incluido en el entorno de desarrollo de Arduino y comprobar que no exista ningún problema a la hora de la carga, así como que el Arduino lo ejecute. El programa cambia el voltaje de salida del pin número 13 de la placa de desarrollo Arduino, haciendo que el LED conectado a ese puerto parpadee. Es el programa más simple disponible para este tipo de microcontroladores, y es el equivalente al *Hello World* de cualquier programa computacional.

La segunda prueba es verificar que las entradas importantes del Arduino funcionen correctamente. Esta prueba consiste en una subprueba por cada una de las entradas analógicas utilizadas durante el proyecto, es decir, se necesita realizar 5 pruebas distintas. El programa de la prueba utiliza el convertidor analógico digital de estas entradas, cambia manualmente la conexión de los puertos a 5V o tierra y comprueba que lo esté midiendo correctamente. Para Arduino, 5 V tiene un valor de 255, mientras que 0 V tiene un valor de 0.

3.4.2 Pruebas para módulo GPS

Las pruebas del GPS las podemos dividir en cuatro. La primera es verificar que el GPS obtiene señal, la segunda es que el GPS envíe datos correctos a través de la comunicación serial, la tercera es que la librería TinyGPS+ esté haciendo la conversión de datos de manera correcta, y la cuarta es comprobar la precisión del dispositivo.

La primera prueba es visual. Una vez conectado y encendido el dispositivo, el LED que está en la etiqueta *Fix* deberá dejar de parpadear y deberá estar encendido. Esta prueba nos ayuda a saber si el área en la que nos encontramos no tiene interferencias importantes que impidan que nuestro módulo detecte y procese la señal GPS. En caso de que el módulo no muestre este comportamiento, la solución común es buscar un espacio con vista directa al cielo.

La segunda prueba comprueba que la comunicación serial entre el Arduino y el módulo GPS funciona correctamente. El programa inicia la comunicación Serial del Arduino con nuestra computadora, e inicia la comunicación serial de software en los puertos que determinemos. Para el caso del GPS, estos puertos suelen ser 8 (Recepción) y 7 (Transmisión). Además, se debe configurar un *Baud Rate* de 9600 para ambas conexiones. El resultado esperado es que al subir el programa al Arduino e iniciar el monitor serial, se observe la transmisión de código NMEA por parte del módulo GPS. Esto debe funcionar sin importar si el módulo puede fijar su posición o no.

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(8,7); // RX, TX
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
}
void loop() {
  if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}

```

La tercera prueba busca verificar el funcionamiento del *parsing* de la librería TinyGPS+. Esta prueba se debe realizar cuando el módulo GPS ha logrado fijar su posición. El programa establece la comunicación serial entre el Arduino y la computadora, inicializa los objetos GPS de la librería TinyGPS+ y envía la latitud, longitud, hora y fecha recopiladas de la información de posición satelital. Esto se puede verificar en el monitor serial del entorno de desarrollo Arduino.

La cuarta prueba con relación al módulo GPS es verificar la precisión de las coordenadas obtenidas comparando los datos extraídos durante la tercera prueba con nuestra posición real. La metodología consiste en obtener las coordenadas con 4 cifras significativas e insertarlas en la aplicación de *Google Maps*. En esta aplicación podemos definir qué tan cerca nos muestra los datos del GPS a comparación de nuestra posición real.

3.4.3 Pruebas para módulo GSM

De forma similar al módulo GPS, para nuestro módulo SIM900 existen tres pruebas que nos permiten saber si el dispositivo está funcionando correctamente. La primera de ellas es

DESCRIPCIÓN Y DESARROLLO

verificar que nuestro módulo se registra correctamente a la red de telefonía celular, la segunda prueba muestra que el Arduino puede comunicarse con el módulo celular, y la tercera muestra que el módulo es capaz de realizar llamadas.

La primera prueba es visual. Una vez conectado y encendido el dispositivo, se debe verificar que el LED de *PWR* del módulo celular esté encendido de manera constante y que además el LED *NET* este parpadeando cada 3 segundos. En caso de que esto no suceda puede ser por dos probables razones

- El módulo celular no se encendió de manera independiente. Para solucionar esto se debe de presionar el botón de *PWRKEY* y esperar a que el LED *NET* comience a parpadear cada 3 segundos.
- La energía proporcionada no es suficiente. Si la fuente externa no suministra la corriente necesaria, el módulo celular se mantendrá en el circuito de encendido.

La segunda prueba consiste en establecer la comunicación serial del Arduino con el módulo celular, y el envío de un código AT para corroborar el correcto funcionamiento del dispositivo. Para esto, se debe cargar un programa el cual iniciará la comunicación serial con la computadora y después iniciará una comunicación serial tipo *software* con el módulo celular, usando los puertos 2 (Recepción) y 3 (Transmisión). Además, se debe configurar un *Baud Rate* de 9600 para ambas conexiones. El resultado esperado es que al subir el programa al Arduino e iniciar el monitor serial, se envíe el comando AT y se muestre en el monitor serial. Se espera como respuesta que el módulo responda con una repetición de nuestro comando además de un *OK*. Esto debe funcionar sin importar si el módulo celular está conectado a la red o no.

```

#include <SoftwareSerial.h>
SoftwareSerial mySerial(2,3); // RX, TX
void setup() {
  // Open serial communications and wait for port to open:
  Serial.begin(9600);
  // set the data rate for the SoftwareSerial port
  mySerial.begin(9600);
}
void loop() {
  if (mySerial.available()) {
    Serial.write(mySerial.read());
  }
  if (Serial.available()) {
    mySerial.write(Serial.read());
  }
}
}

```

La tercera prueba consiste en hacer que el módulo genere una llamada. Para ello, las pruebas anteriormente descritas deben haber funcionado. El programa que se debe cargar al Arduino consiste en la configuración de una comunicación serial con la computadora, además de iniciar la comunicación serial con el Arduino con el módulo celular. Se envía un comando AT con un número celular predefinido y se verifica que se reciba una llamada.

PRUEBAS Y RESULTADOS

Resultados esperados

- Enviar un mensaje de texto en menos de 3 minutos con una probabilidad de éxito de al menos un 90 % en todos los casos de prueba de accidentes.
- Detección efectiva de colisiones y velocidad de impacto. Delta V (dV).
- Poder cambiar datos de usuario.
- Poder realizar una llamada a un número predeterminado.

4.1 Resultados esperados

Como se ha comentado el presente proyecto pretende dar solución a un problema actual en el ámbito del transporte desarrollando un modelo funcional sin las características necesarias para ser un dispositivo comercial. Esto no implica que esta prueba de concepto esté exenta de las funciones mínimas requeridas para dar solución a la problemática a la que nos enfrentamos.

Para ello se establecen algunos parámetros fundamentales que ayudarán a determinar si los resultados obtenidos cumplen satisfactoriamente con los propósitos de este proyecto.

El primero de estos parámetros es la correcta detección y aviso de la existencia de un percance vehicular. Este problema puede dividirse en dos puntos: correcta detección de un accidente y envío de una alerta a través de un mensaje de texto sms.

A continuación, se verifica la información contenida en el mensaje de texto recibido por los servicios de emergencias. Esta información debe contener al menos el nombre de la persona accidentada la cual deberá tener una extensión de 160 caracteres por cuestiones de la longitud del mensaje.

El tercer parámetro a evaluar será la información relacionada con la ubicación geográfica del accidente compuesta por latitud y longitud. Estos datos contenidos en el mensaje de texto deberán tener un formato previamente establecido de al menos dos cifras significativas.

El último parámetro a verificar será la precisión en la diferencia de velocidades, así como la determinación de los sensores impactados.

Para la evaluación de estos cuatro parámetros ya descritos llevaremos a cabo un método de ensayo donde tomaremos la velocidad y el número de sensores activados como variables. El procedimiento de este método estará dado por cinco velocidades y cuatro posiciones de sensores previamente establecidas. Esto generará 20 resultados de pruebas con los que se podrá determinar el éxito o fracaso de los parámetros a evaluar.

Tabla 4.1.1 Resultados Esperados

Categoría	Pregunta	Frecuencia
Mínimo	¿Se detectó accidente?	18/20
	¿Se dio instrucción de envío de SMS?	18/20
	¿Se recibió SMS?	18/20
SMS con nombre	¿Se recibió el nombre?	15/20
	Cambio correcto de datos	4/5
GPS	¿Tiene señal?	16/20
	¿Se reciben los datos correctamente?	16/20
	Localización recibida	
	Localización real	
	Diferencia en metros	
	¿Diferencia aceptable?	10 m
Sensores y Δv	Número de sensores recibidos	
	Número de sensores reales	
	Diferencia	≤ 1

PRUEBAS Y RESULTADOS

	Velocidad real	
	Velocidad marcada	85 % de precisión con real
	Δv marcado/enviado	
Llamada	¿Se enlazó la llamada?	3/5

Cambio temporal de datos del conductor

Conocer la información de quién conduce el vehículo es una parte primordial de este proyecto si tenemos en cuenta que el SMS que se envía en caso de accidente a los servicios de emergencias incluye datos como nombre y apellidos, número de licencia, ubicación, sensores impactados, variación en la velocidad y tipo de sangre. Esta información estará guardada de forma permanente en la memoria interna del Arduino, sin embargo, una situación común podrá ser el cambio de conductor de manera temporal. Para este tipo de situaciones se implementan las funciones de recepción e interpretación de SMS por parte del módulo GSM y del Arduino respectivamente, es decir, el nuevo usuario podrá enviar un mensaje de texto al dispositivo lo que temporalmente modificará los datos que serán enviados a emergencias en caso de accidente.

Formato:

@Nombre y Apellidos;#Número;\$NúmeroDeLicencia;&TipoDeSangre;

Distribución de caracteres (hasta 160):

@40#10\$10&3;

Total de caracteres recibidos por el dispositivo: 68

Ejemplo:

@JUAN PEREZ PEREZ;#5500112233;\$C11110000;&AB+;

4.2 Resultados obtenidos

A continuación se enumeran los casos relacionados con 20 pruebas llevadas a cabo en una misma ubicación y bajo las mismas condiciones climáticas y del pavimento.

Desde un punto de vista práctico, el cambio en la ubicación no afectará el objetivo de este proyecto pues la probabilidad de tener señal en el GPS en este escenario será la misma que en otras zonas donde se pudiesen llevar a cabo estas pruebas. Es cierto que las condiciones climáticas, tales como lluvia, nieve, neblina, o si es de día o de noche, influyen de manera importante en la visibilidad del conductor, así como en la correcta recepción de los datos del GPS y por lo tanto influyen en la gravedad del accidente. Sin embargo, para la etapa de desarrollo en la que nos encontramos, la variación de las condiciones climáticas y del pavimento no deben afectar de manera significativa el funcionamiento del dispositivo.

Las primeras cinco pruebas realizadas contendrán los datos introducidos manualmente en el entorno de desarrollo de Arduino. Las pruebas números 6, 10, 11 y 17 se realizarán partiendo de un cambio de datos del usuario de manera manual a través de un mensaje de texto SMS que se enviará al dispositivo al iniciar el programa, es decir, se realizarán cuatro cambios en los datos.

Para la comunicación a través de una llamada se empleará la misma lógica, es decir, sólo en cinco de los 20 casos se realizarán llamadas. Estas serán en las pruebas 4, 7, 10, 14 y 18.

Tabla 4.2.1 Resultados Obtenidos

Categoría	Pregunta	Esperado	Obtenido
Mínimo	¿El dispositivo detectó accidente?	18/20	20/20
	¿Se dio instrucción de envío de SMS?	18/20	20/20
	¿Se recibió SMS?	18/20	19/20
SMS con nombre	¿Se recibieron los datos?	3/4	4/4

PRUEBAS Y RESULTADOS

	Cambio correcto de datos	3/4	4/4
	Nombre correcto	3/4	4/4
GPS	¿Tiene señal?	16/20	20/20
	¿Se reciben los datos correctamente?	16/20	19/20
	Promedio diferencia en metros	≤ 10 m	6.36 m
	¿Diferencia aceptable? (10 m)	14/20	15/20
Sensores y Δv	Diferencia entre sensores reales y sensores recibidos	18/20	17/20
	Δv recibido (85 %)	12/15	10/15
Llamada	¿Se enlazó la llamada?	3/5	4/5

En lo que al mínimo necesario se refiere, el dispositivo se comportó mejor de lo que se esperaba. En dos de los tres conceptos esperados se logró un 100 % de funcionamiento, mientras que en el último de estos no se recibió el mensaje en una ocasión. Esto nos da a entender que en su forma más básica, la detección y aviso de un accidente a través de SMS, el dispositivo funciona con una alta efectividad. El error en la recepción del SMS podría estar dado por fallas en la red del proveedor de servicios de comunicaciones, fallas en el dispositivo celular con el que se estaban realizando las pruebas o por fallas del módulo GSM del dispositivo en cuestión. Nos es imposible determinar la causa, pero dados los resultados previos y debido a que la instrucción de envío del SMS fue dada, consideramos que fue una falla en la red.

En las pruebas realizadas en cuanto al cambio de datos de usuario, el dispositivo se comportó de manera precisa. En todas las pruebas realizadas hubo una recepción del mensaje que se envió desde nuestro celular, y en todas las ocasiones se procesó y envió el nombre modificado. Tomando en cuenta la experiencia de la sección previa, podemos decir que si se

realizara un mayor número de pruebas en esta sección, la infalibilidad sólo quedaría condicionada por el correcto funcionamiento de la red celular.

La sección de las pruebas realizadas sobre el GPS también superó nuestras expectativas, pero no fueron tan exitosas como las secciones anteriores. La distancia promedio de diferencia entre la ubicación marcada del accidente y la ubicación real recibida en promedio fue de 6.36 metros, y en 15 de las 20 pruebas esta diferencia no superó los 10 metros de diferencia. Esto nos indica que la precisión de la ubicación es lo suficientemente acertada como para que en una situación real el dispositivo proporcione una localización desde la cual los servicios de emergencia, por ejemplo una ambulancia, podrán divisar al accidentado. Además, esta diferencia puede deberse en gran parte a la forma de realizar los experimentos, dado que la ubicación marcada como accidente fue previamente establecida, y el dispositivo se accionaba cuando el vehículo de pruebas pasaba sobre él. Esta precisión podría mejorar si las pruebas se realizaran con un dispositivo GPS adicional que marcara el lugar preciso en donde se activaron los sensores del dispositivo.

En cuanto a las mediciones de los sensores y la diferencia de velocidad, los resultados obtenidos fueron inferiores a los esperados. El número de sensores activados recibidos varió en 3 ocasiones de las 20 pruebas realizadas, mientras que la velocidad recibida se encontró dentro del rango aceptado en 10 de las 15 pruebas (recordemos que las pruebas realizadas mientras el vehículo de pruebas estaba estacionario no tienen un rango de velocidad adecuado). La diferencia de los sensores se puede deber a dos posibles razones, la primera de ellas el factor humano, es decir, que el operador encargado de presionar los botones pudo haber cometido el error de no presionar el botón de los sensores correctamente, o por alguna falla en el circuito de los sensores. La primera causa es mucho más probable, por lo que si se desea aumentar este porcentaje en las pruebas, la activación de los sensores se debería

PRUEBAS Y RESULTADOS

automatizar también. Por otro lado, el error en la velocidad registrada puede deberse a tres distintas razones. La primera es debido a la diferencia de la velocidad real con el velocímetro del vehículo de pruebas, que puede ser poco preciso, la segunda a la frecuencia de actualización del cálculo de velocidades del módulo GPS del dispositivo, ya que recordemos que actualiza la ubicación cada un segundo, pero el cálculo de velocidades parece llevar un retraso más amplio, y finalmente la tercera el error humano, ya que dependíamos de un operador que daba aviso de cuando la velocidad de la prueba era alcanzada. Para mejorar estos rangos, se sugiere realizar las pruebas con un vehículo con un velocímetro digital y en una pista vehicular privada para poder mantener la velocidad de la prueba sin interferencia del entorno por un tiempo mucho mayor.

Finalmente, para las pruebas de las llamadas, detectamos que se logró enlazar esta llamada en 4 de las 5 pruebas en las que se hizo este intento. Se sospecha que la prueba errónea se debe a posibles fallas en la red pues en las 5 ocasiones se recibió el mensaje SMS, indicando que el módulo GSM estaba trabajando de manera correcta. Además, para enlazar una llamada, la red necesita mucha mayor capacidad que para un mensaje, por lo que si se realizara un número mayor de pruebas, se esperaría que el número de llamadas totales enlazadas fuera mucho menor al número de mensajes recibidos.

CONCLUSIONES

5.1 Conclusiones generales

El desarrollo de la prueba de concepto resultó exitoso, logrando superar en algunos rubros las expectativas que se tenían sobre el dispositivo. Esto, por supuesto, no implica que el dispositivo esté listo para salir al mercado, pero sí ofrece una buena perspectiva sobre cómo iniciar el proceso de un prototipo.

En general y a manera de resumen, se utilizó un Arduino UNO como microprocesador, un módulo SIM 900 para la conexión GSM, un módulo *Tinysine GPS Logger Shield V2* para la geolocalización y *push buttons* como simuladores de detectores de impacto o sensores.

El dispositivo tiene la capacidad de:

- Cambiar los datos almacenados al iniciar a través de la recepción de un mensaje SMS.
- Detectar la velocidad a la que el vehículo se está desplazando y determina la velocidad máxima detectada en el momento del impacto.
- Detectar cuándo se han activado los sensores de impacto, y determinar el número de estos que fueron estimulados.
- Consolida la información obtenida de los sensores y del módulo GPS junto con la información guardada sobre el usuario.
- Envía un mensaje SMS con toda la información previamente descrita a un número previamente establecido.
- Detecta si el usuario desea establecer una llamada y la genera en caso de ser necesario.

CONCLUSIONES

Las herramientas, tales como los módulos o el microcontrolador, utilizadas en este proyecto resultaron adecuadas, pues se logró la función que se esperaba de ellas sin que sus características causaran conflicto con el funcionamiento general del dispositivo.

5.2 Planes a futuro

El continuo perfeccionamiento de este dispositivo es un claro objetivo pretendido para este proyecto. Habiendo conseguido que se detecte un accidente correctamente y que se envíe de igual manera un mensaje de texto y/o se enlace una llamada a los servicios de emergencias, consideramos es momento de plantearnos ciertos planes a futuro que mejorarán gradualmente sus funcionalidades y permitirán una explotación más completa del dispositivo.

En la última década el sector tecnológico se ha visto impulsado gracias al desarrollo de nuevos productos “inteligentes” tales como *smartphones*, *tablets*, *smart TVs*, entre otros. Esto, a su vez, ha abierto las puertas a la creación de aplicaciones digitales con innumerables usos inundando así las tiendas electrónicas como *Google Play Store* o *Apps Store* de Apple. Apoyados en esta nueva tendencia que cada día cuenta con más usuarios alrededor del mundo, consideramos importante el desarrollo de una aplicación digital o *app*, como comúnmente se les conoce, la cual mejore la interacción entre el usuario y el dispositivo y además le añada nuevas funcionalidades al mismo.

Para lograr esto es evidente que nuestro módulo GPRS no será suficiente en cuanto a los requisitos necesarios para realizar una conexión satisfactoria a internet, por lo que será imprescindible migrar nuestro sistema a módulos mejorados tales como el módulo *Skywire 4G* de *NimbeLink* o aquellos, actualmente aún en desarrollo, que utilicen tarjetas eSIM o SIM virtual.

Habiendo discutido esto, el nuevo modelo de nuestro dispositivo podrá conectarse fácilmente a la gran red a través de la aplicación celular, lo que supondrá algunas innovaciones en lo que a posibilidades de explotación se refiere. El siguiente listado enumera algunas de estas nuevas ventajas a implementar en el futuro a través de una *app*:

CONCLUSIONES

- Desarrollo de un sistema de mensajería.
- Localización en caso de robo.
- Seguimiento en tiempo real de la trayectoria del vehículo.
- Actualización en tiempo real de plataformas como *Waze* o *Google Maps* de usuarios accidentados.
- Inclusión de nueva información en mensajes.

Ya que la información podrá ser enviada a través de internet en un futuro, consideramos conveniente incrementar la longitud del mensaje con datos relacionados con el usuario, por ejemplo, tipo de seguro médico y aseguradora.

Una de las grandes ventajas de estos dispositivos móviles con conexión a internet es la comunicación entre ellos. Esto permitirá que los dispositivos de dos o más vehículos se comuniquen entre sí en situaciones donde se requiera, por ejemplo, en un accidente, pudiendo determinar así la velocidad, dirección y otros datos del recorrido de cada vehículo y comparándola, lo que podría facilitar el trabajo de las aseguradoras y los servicios de emergencias.

REFERENCIAS BIBLIOGRÁFICAS

- Fourth Generation Mobile Communication. The Path to Superfast Connectivity; Curwen, Peter; Whalley, Jason; Editorial Springer; 2013; ISBN 978-3-319-022109 (eBook).
- GPRS. General Packet Radio Service; Bates, Regis J.; Editorial McGraw-Hill; 2002; ISBN 978-0-071-39413-0 (eBook).
- 4G: LTE/LTE-Advanced for Mobile Broadband; Dahlman, Erik; Parkvall, Stefan; Sköld, Johan; Editorial Elsevier; Segunda Edición; 2014; ISBN 978-0-12-419985-9.
- LTE and the Evolution to 4G Wireless. Design and Measurement Challenges; Agilent Technologies, Inc.; Editorial John Wiley & Sons, Inc.; Segunda Edición por Moray Rumney; 2013; ISBN 978-1-119-96257-1.
- GPRS and 3G Wireless Applications; Anderson, Christoffer; Editorial John Wiley & Sons, Inc.; Primera Edición; 2002; ISBN: 978-0-471-18975-6.
- Artículo: Low Cost Embedded Car Safety and Security Enhancement using ARM; S.Sundar, V.V.R.S.Karthik, Y.Deepak; Revista: International Journal of Applied Engineering Research; Editorial Research India Publications; ISSN 0973-4562 Volumen 9, Número 16; 2014; pp. 3443-3453.
- Artículo: Controlling and Monitoring of Electric Feeders Using GSM Network Technology; Revista: Journal of Electrical and Electronics Engineering; Ahmed Afaz Uddin y otros; Chittagong University of Engineering and Technology (CUET), Bangladesh, Department of Electrical and Electronic Engineering; Volumen 6, Número 2, Octubre de 2013, pp. 5-8.
- GPS. Theory, Algorithms and Applications; Xu, Guochang; Editorial Springer; Segunda Edición; 2007; ISBN 978-3-540-72714-9.

REFERENCIAS BIBLIOGRÁFICAS

- GNSS Remote Sensing. Theory, Methods and Applications; Cardellach, Estel y otros; Editorial Springer; Primera Edición; 2014; ISBN 978-94-007-7482-2 (eBook).
- Tercer Informe sobre la situación de la seguridad vial, México 2013; Cervantes Trejo, Arturo y otros; Secretariado Técnico CONAPRA y Secretaría de Salud; Primera Edición; 2013; ISBN en trámite.
- Análisis de los Sistemas de Urgencias en México. Propuesta de Modelo Operativo; Cervantes Trejo, Arturo y otros; Secretariado Técnico CONAPRA y Secretaría de Salud; Primera Edición; 2013; ISBN en trámite.
- Análisis de los Sistemas de Urgencias en México. Análisis Cualitativo 2008; Cervantes Trejo, Arturo; López Kneeland, Manuel; Secretariado Técnico CONAPRA y Secretaría de Salud; Primera Edición; 2008; sin ISBN.
- Análisis de los Sistemas de Urgencias en México. Análisis Cuantitativo 2008; Cervantes Trejo, Arturo; López Kneeland, Manuel; Secretariado Técnico CONAPRA y Secretaría de Salud; Primera Edición; 2008; sin ISBN.
- Caracterización de las defunciones por accidentes de transporte en la República Mexicana Cervantes Trejo, Arturo; Hernández Polo, Yolanda; Secretariado Técnico CONAPRA y Secretaría de Salud; Primera Edición; sin ISBN.
- Global Status Report on Road Safety 2013: supporting a decade of action; Organización Mundial de la Salud; 2013; ISBN 978-92-4-156456-4.
- Informe sobre el Estado de la Seguridad Vial en la Región de las Américas; Organización Panamericana de la Salud; 2009; ISBN 978-92-75-33069-2.

- Programmable Microcontrollers with Applications: MSP430 LaunchPad with CCS and Grace; Ünsalan, Cem; Gürham, H. Deniz; Editorial McGraw-Hill; Primera Edición; 2014; ISBN 978-0-07-183004-1.

Otras referencias

- Causes of death 2008: data sources and methods; Departamento de Estadística e Informática Sanitarias; Organización Mundial de la Salud; Ginebra; Abril 2011.
- Estadística de defunciones generales. Descripción de la base de datos; Instituto Nacional de Estadística y Geografía; 2014.
- Reglamento de Tránsito del Distrito Federal y Aviso; Gaceta Oficial del Distrito Federal; Número 156 Bis; 17 de agosto de 2015; Administración Pública del Distrito Federal, Jefatura de Gobierno.
- Artículo: “Para conducir moto, elija el mejor casco.” Salomón Rodríguez; El Economista, 10 de enero de 2013.
- <http://gpsworld.com/tag/buyers-guide/>
- <http://simcom.ee/modules/gsm-gprs/sim900/>
- <https://www.arduino.cc/en/main/arduinoBoardUno>

Protocolos de comunicación serial

- <https://learn.sparkfun.com/tutorials/serial-communication>
- <http://www.ti.com/lit/ug/sprugp1/sprugp1.pdf>
- <http://www.eeherald.com/section/design-guide/esmod7.html>
- <https://learn.sparkfun.com/tutorials/serial-peripheral-interface-spi>

REFERENCIAS BIBLIOGRÁFICAS

GNSS

- http://www.unoosa.org/pdf/icg/2013/Ed_GNSS_eBook.pdf
- <http://techdigest.jhuapl.edu/TD/td1901/guier.pdf>
- <http://www.gps.gov/systems/gps/space/>
- <https://www.glonass-iac.ru/en/GLONASS/>
- http://www.colorado.edu/geography/gcraft/notes/gps/gps_f.html
- <http://www.globmaritime.com/technical-articles/marine-navigation/general-concepts/9622-trilateration-traverse-and-vertical-surveying.html>
- http://www.rand.org/content/dam/rand/pubs/monograph_reports/MR614/MR614.apa.pdf
- <http://www.gps.gov/technical/ps/2008-SPS-performance-standard.pdf>

ANEXOS

Código Final

```
//Declaracion de librerias
#include <TinyGPS++.h>
#include <SoftwareSerial.h>

//Declaracion de variables
static const int RXPin = 8, TXPin = 7;
TinyGPSPlus gps;
SoftwareSerial ss(RXPin, TXPin);
SoftwareSerial ssgsm(2,3);
boolean flag;
String mensaje;
String latitud;
String longitud;
String nombre = "nombre", numero, sangre, licencia;
String text;
String sensoresActivadosStr;
char apoyo[100];
double primeraVelocidad, segundaVelocidad;
double speedLog[20];
boolean botonSMS,banderaApoyo,banderaLoopPrincipal;
int contadorEsperaSMS,contadorInicio;

int sensor1,sensor2,sensor3,sensor4,sensoresActivados,j;
double i;

// variables will change:
int buttonState = 0;          // variable for reading the pushbutton
status

void setup() {
  // initialize the LED pin as an output:
  //pinMode(ledPin, OUTPUT);
  // initialize the pushbutton pin as an input:
  pinMode(A0,INPUT);
  pinMode(A1,INPUT);
  pinMode(A2,INPUT);
  pinMode(A3,INPUT);
  pinMode(A5,INPUT);

  Serial.begin(9600);
```

ANEXOS

```
ss.begin(9600);
delay(1000);
Serial.println("Antes de mandar mensaje");
sensor1=0;
sensor2=0;
sensor3=0;
sensor4=0;
flag = true;
botonSMS = false;
banderaApoyo = true;
banderaLoopPrincipal = false;
//ssgsm.begin(9600);
}
```

```
void loop(){

  if(digitalRead(A5)&&banderaApoyo){
    botonSMS = true;
    Serial.println("se apreto el boton");
    banderaApoyo = false;
    receiveSMSConfig();
    ss.end();
    delay(1000);
    ssgsm.begin(9600);
    delay(500);
  }

  if(contadorInicio>1000){
    banderaApoyo = false;
    banderaLoopPrincipal = true;
  }
  else{
    contadorInicio++;
    delay(10);
    Serial.println(contadorInicio);
  }

  while(botonSMS){
    delay(50);
    contadorEsperaSMS++;
    Serial.println(contadorEsperaSMS);
    if (ssgsm.available()){
      text = ssgsm.readString();
      Serial.println(text);
    }
  }
}
```

```

        nombre =
text.substring(text.indexOf("@")+1,text.indexOf("#"));
        numero =
text.substring(text.indexOf("#")+1,text.indexOf("&"));
        sangre =
text.substring(text.indexOf("&")+1,text.indexOf("$"));
        licencia =
text.substring(text.indexOf("$")+1,text.indexOf(";"));
        Serial.println(nombre);
        Serial.println(numero);
        Serial.println(sangre);
        Serial.println(licencia);
    }

    if(contadorEsperaSMS>500){
        botonSMS = false;
        Serial.println("Tiempo de espera agotado");
        banderaLoopPrincipal = true;
        ssgsm.end();
        ss.begin(9600);
    }
    if(contadorEsperaSMS>40 && digitalRead(A5)){
        botonSMS = false;
        Serial.println("Fin de ciclo forzado");
        banderaLoopPrincipal = true;
        ssgsm.end();
        ss.begin(9600);
    }
}

/*Lectura de SMS*/
while(ss.available()>0 && banderaLoopPrincipal){ //Detecta que
se reciban caracteres del GPS
    if(gps.encode(ss.read())){ //Codifica los caracteres del GPS
        updateSpeedlog();
        if(isAccident()){
            getSensors(2400);
            printSpeedlog();
            prepareSMSStr();
            sendAccidentSMS();
            generateCall();
            Serial.println("Llamando");
            hangupCall();
        }
    }
}

```

ANEXOS

```
    }
    Serial.print(banderaLoopPrincipal);
    Serial.print("      ");
    Serial.println(ss.available());
}

void updateSpeedlog(){
    for(j=0;j<19;j++){
        primeraVelocidad = speedLog[j+1];
        speedLog[j] = primeraVelocidad;
    }
    speedLog[19] = gps.speed.kmph();
    primeraVelocidad=gps.speed.kmph();
    Serial.println(isAccident());
}

boolean isAccident(){
    if(digitalRead(A0)){
        sensor1 =1;
    }
    if(digitalRead(A1)){
        sensor2 =1;
    }
    if(digitalRead(A2)){
        sensor3 =1;
    }
    if(digitalRead(A3)){
        sensor4 =1;
    }
    if(sensor1||sensor2||sensor3||sensor4){
        return true;
    } else {
        return false;
    }
}

void getSensors(int waitingTime){
    while(i<waitingTime){
        if(digitalRead(A0)){
            sensor1 =1;
        }
        if(digitalRead(A1)){
            sensor2 =1;
        }
        if(digitalRead(A2)){
            sensor3 =1;
        }
    }
}
```

```

    }
    if(digitalRead(A3)){
        sensor4 =1;
    }
    i++;
}
Serial.println(sensor1);
Serial.println(sensor2);
Serial.println(sensor3);
Serial.println(sensor4);
sensoresActivados=sensor1+sensor2+sensor3+sensor4;
}

void printSpeedlog(){
    for(j=0;j<20;j++){
        Serial.print(speedLog[j]);
        Serial.print(" ");
    }
}

void prepareSMSStr(){
    latitud = String(gps.location.lat(),6);
    longitud = String(gps.location.lng(),6);
    sensoresActivadosStr = String(sensoresActivados);
    mensaje = "Accidente "+nombre+ String(" Lat: ") + latitud +
String(" Long: ") + longitud+String(" Sensores: ") +
sensoresActivadosStr + String(" DeltaV: ") + String(primeravelocidad-
segundaVelocidad);
}

void sendAccidentSMS(){
    ss.end();
    delay(1000);
    ssgsm.begin(9600);
    mensaje.toCharArray(apoyo,100);
    ssgsm.print("AT+CMGS=\"");
    ssgsm.print("5520021909");
    ssgsm.println("\");
    delay(1000);
    ssgsm.print(mensaje);
    ssgsm.write(0x1a);
    ssgsm.println();

}

void generateCall(){

```

ANEXOS

```
    delay(10000);
    while(digitalRead(A5)==0){
        Serial.println("esperando llamada");
    }
    ssgsm.println("ATD5520021909;");
}

void hangupCall(){
    delay(4000);
    while(digitalRead(A5)){
        Serial.println("esperando a colgar");
    }
    //call.HangUp();
    ssgsm.println();
}

void receiveSMSConfig()
{
    delay(500);
    ssgsm.print("ATE1\r\n");
    delay(200);
    ssgsm.print("AT\r\n");
    delay(200);
    ssgsm.print("AT+CMGF=1\r\n");
    delay(200);
    ssgsm.print("AT+CNMI=1,2,0,0,0\r\n");
    delay(200);
}
```

Desarrollo de Pruebas

Categoría Mínimo:

Prueba	¿El dispositivo detectó accidente?	¿Se dio instrucción de envío de SMS?	¿Se recibió SMS?	Tiempo de recepción [segundos]
1	Sí	Sí	Sí	6
2	Sí	Sí	Sí	4
3	Sí	Sí	Sí	7
4	Sí	Sí	Sí	9
5	Sí	Sí	Sí	10
6	Sí	Sí	Sí	7
7	Sí	Sí	Sí	4
8	Sí	Sí	Sí	6
9	Sí	Sí	Sí	7
10	Sí	Sí	Sí	4
11	Sí	Sí	Sí	7
12	Sí	Sí	Sí	5
13	Sí	Sí	Sí	7
14	Sí	Sí	No	-
15	Sí	Sí	Sí	4
16	Sí	Sí	Sí	8
17	Sí	Sí	Sí	8
18	Sí	Sí	Sí	6
19	Sí	Sí	Sí	7
20	Sí	Sí	Sí	5

ANEXOS

Categoría SMS con Nombre:

Pruebas	¿Se recibieron los datos?	Cambio correcto de datos	Nombre correcto
6	Sí	Sí	Sí
10	Sí	Sí	Sí
11	Sí	Sí	Sí
17	Sí	Sí	Sí

Categoría GPS:

Puntos de referencia: [19.376134/-99.123190] “Punto A”; [19.377247/-99.123154] “Punto B”; [19.376911/-99.123159] “Punto C”; [19.376603 -99.123177] “Punto D”.

Prueba	¿Tiene señal?	¿Se reciben los datos correctamente?	Localización real del accidente	Localización recibida	Δd	¿ $\pm 10m$?
1	Sí	Sí	19.376134 -99.123190	19.376093 -99.123181	5	Sí
2	Sí	Sí	19.376134 -99.123190	19.376080 -99.123212	6	Sí
3	Sí	Sí	19.376134 -99.123190	19.376092 -99.123208	5	Sí
4	Sí	Sí	19.376911 -99.123159	19.377026 -99.123177	13	No
5	Sí	Sí	19.376911 -99.123159	19.377043 -99.123161	15	No
6	Sí	Sí	19.376134 -99.123190	19.376142 -99.123214	0	Sí
7	Sí	Sí	19.377247 -99.123154	19.377246 -99.123180	0	Sí
8	Sí	Sí	19.376603 -99.123177	19.376566 -99.123260	4	Sí
9	Sí	Sí	19.376603 -99.123177	19.376543 -99.123108	6	Sí
10	Sí	Sí	19.376603 -99.123177	19.376640 -99.123222	4	Sí
11	Sí	Sí	19.376134 -99.123190	19.376218 -99.123189	10	Sí
12	Sí	Sí	19.376134 -99.123190	19.376199 -99.123216	7	Sí
13	Sí	Sí	19.377247 -99.123154	19.377254 -99.123207	0	Sí
14	Sí	No	19.377247 -99.123154	-	-	No

ANEXOS

15	Sí	Sí	19.376603 -99.123177	19.376577 -99.123192	3	Sí
16	Sí	Sí	19.376134 -99.123190	19.376174 -99.123163	5	Sí
17	Sí	Sí	19.377247 -99.123154	19.377327 -99.123133	9	Sí
18	Sí	Sí	19.376603 -99.123177	19.376566 -99.123177	4	Sí
19	Sí	Sí	19.376911 -99.123159	19.376780 -99.123085	14	No
20	Sí	Sí	19.376911 -99.123159	19.376814 -99.123169	11	No

Categoría Sensores y Δv

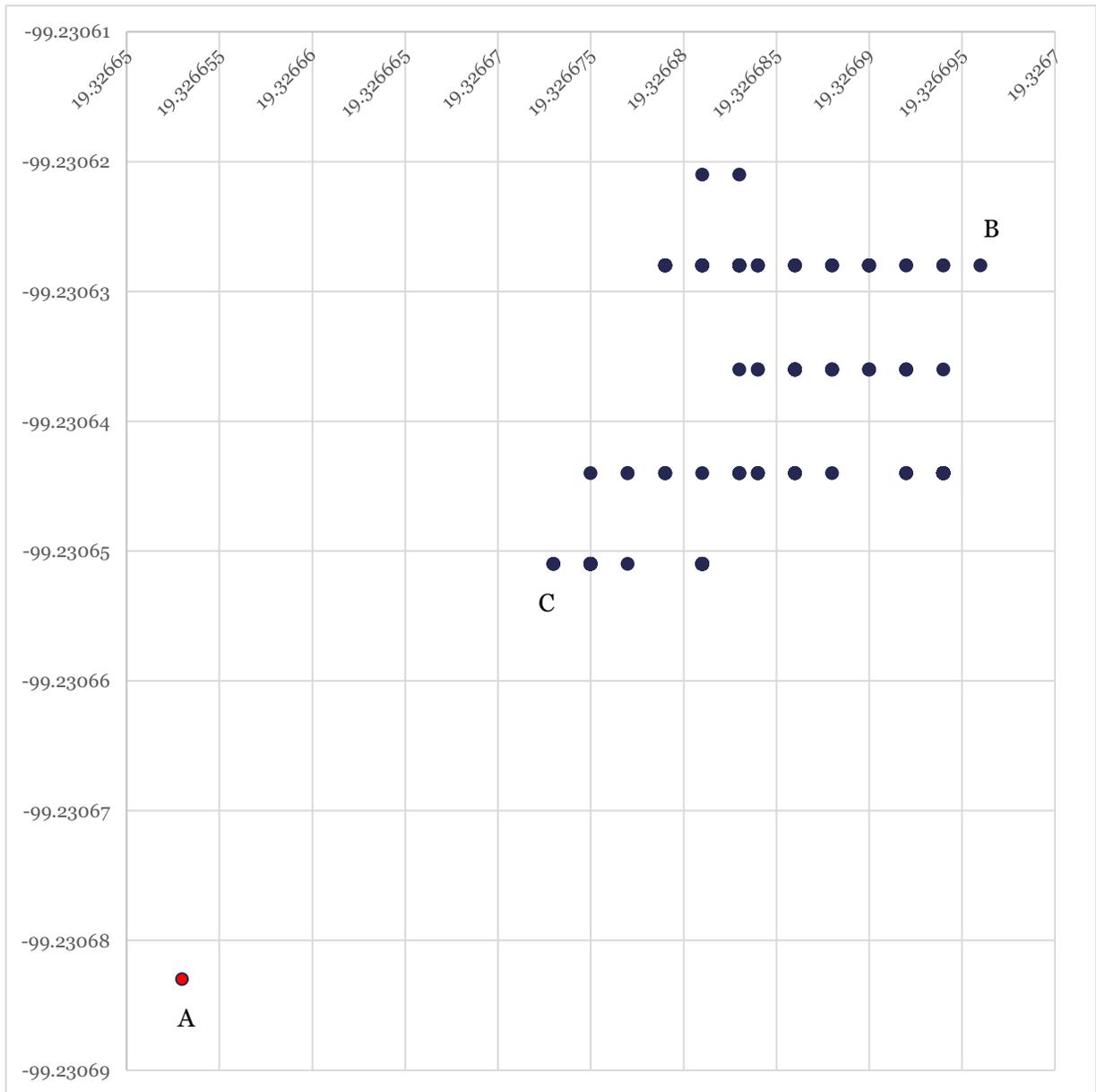
Prueba	Número de sensores recibidos	Número de sensores reales	Velocidad real [km/h]	Velocidad marcada [km/h]	Diferencia (Módulo)	¿%? Ideal 85
1	1	1	0	2.02	2.02	-
2	2	1	10	3.85	6.15	38.5
3	1	1	20	15.98	4.02	79.9
4	2	1	25	21.89	3.11	87.56
5	1	1	35	32.19	2.81	91.97
6	2	2	0	0.46	0.46	-
7	2	2	10	9.74	0.26	97.4
8	2	2	20	17.5	2.5	87.5
9	2	2	25	26.15	1.15	95.4
10	2	2	35	30.15	4.85	86.14
11	3	3	0	0.09	0.09	-
12	3	3	10	8.96	1.04	89.6
13	3	3	20	17.56	2.44	87.8
14	-	3	25	-	-	-
15	3	3	35	28.52	6.48	81.4
16	4	4	0	0.26	0.26	-
17	4	4	10	12.28	2.28	77.2
18	4	4	20	20.76	0.76	96.2
19	4	4	25	23.67	1.33	94.68
20	4	4	35	32.02	2.98	91.48

ANEXOS

Categoría Llamada:

Pruebas	¿Se enlazó la llamada?
4	Sí
7	Sí
10	Sí
14	No
18	Sí

Pruebas de precisión y exactitud del Módulo GPS



Distancia del Punto A al punto más lejano (Punto B) = 7.5 m aproximadamente.

Distancia del Punto B al Punto C = 3.5 m aproximadamente.

ANEXOS

Tomando en cuenta que el límite establecido para determinar como visible un accidente en este proyecto es de 10 metros, el presente experimento puede considerarse como exitoso y el dispositivo como preciso, mas no exacto.

Glosario

- **3GPP** *Third-Generation Partnership Project*
- **ADC** *Analog-to-Digital Converter*
- **ALU** *Arithmetic Logic Unit*
- **AMPS** *Advanced Mobile Phone System*
- **ARQ** *Automatic Repeat Request*
- **ASCII** *American Standard Code for Information Interchange*
- **AUC** *Authentication Centre*
- **BSC** *Base Station Controller*
- **BSS** *Base Station Subsystem*
- **BTS** *Base Transceiver Station*
- **CCITT** *Comité Consultivo Internacional Telegráfico y Telefónico*
- **CENAPRA** *Centro Nacional para la Prevención de Siniestros*
- **CEPT** *Conference of European Posts and Telecommunications*
- **CPU** *Central Processing Unit*
- **DFTS-OFDM** *Discrete Fourier Transform Spread Orthogonal Frequency Division Multiplexing*
- **DSL** *Digital Subscriber Line*
- **EDGE** *Enhanced Data Rates for GSM Evolution*
- **EPROM** *Erasable Programmable Read-Only Memory*
- **eSIM** *electronic Subscriber Identity Module*
- **ETSI** *European Telecommunications Standards Institute*
- **FTP** *File Transfer Protocol*

ANEXOS

- **GLONASS** *GLobal NAVigation Satellite System*
- **GNSS** *Global Navigation Satellite System*
- **GPIO** *General Purpose In/Out*
- **GPRS** *General Packet Radio Services*
- **GPS** *Global Positioning System*
- **GSM** *Groupe Spéciale Mobile (posteriormente Global System for Mobile communications)*
- **HDLC** *High-Level Data Link Control*
- **HLR** *Home Location Register*
- **HTTP** *Hypertext Transfer Protocol*
- **I2C** *Inter-Integrated Circuit*
- **IEEE** *Institute of Electrical and Electronics Engineers*
- **IMSI** *International Mobile Subscriber Identity*
- **IP** *Internet Protocol*
- **ISDN** *Integrated Services Digital Network*
- **ISDN** *Integrated Services Digital Network*
- **ISO** *International Standard Organization*
- **ISP** *Internet service provider*
- **LAN** *Local Area Network*
- **LED** *Light-Emitting Diode*
- **LSB** *Least Significant Bit*
- **LTE** *Long Term Evolution*
- **MAB** *Memory Address Bus*
- **MEO** *Medium Earth Orbit*

- **MISO** *Master Input/Slave Output*
- **MOSI** *Master Output/Slave Input*
- **MS** *Mobile Station*
- **MSB** *Most Significant Bit*
- **MSC** *Mobile Switching Centre*
- **NMEA** *National Marine Electronics Association*
- **NSS** *Network Switching Subsystem*
- **OFDM** *Orthogonal Frequency Division Multiplexing*
- **OMC** *Operation and Maintenance Centre*
- **OMS** *Organización Mundial de la Salud*
- **OSI** *Open Systems Interconnection*
- **PDC** *Personal Digital Cellular*
- **PDU** *Protocol Data Unit*
- **POP3** *Post Office Protocol (versión 3)*
- **PSTN** *Public Switched Telephone Network*
- **RAM** *Random-Access memory*
- **RF** *Radiofrecuencia*
- **RISC** *Reduced Instruction Set Computing*
- **ROM** *Read-Only Memory*
- **SCK** *Serial Clock*
- **SIM** *Subscriber Identity Module*
- **SIM** *Subscriber Identity Module*
- **SMS** *Short Message Service*

ANEXOS

- **SMTP** *Simple Mail Transfer Protocol*
- **SNMP** *Simple Network Management Protocol*
- **SPI** *Serial Peripheral Interface*
- **SS7** *Signaling System number 7*
- **TCP** *Transmission Control Protocol*
- **TDMA** *Time Division Multiple Access*
- **TI** *Texas Instruments*
- **UART** *Universal Asynchronous Receiver-Transmitter*
- **UDP** *User Datagram Protocol*
- **UIT** *Unión Internacional de Telecomunicaciones*
- **UIT-T** *Sector de Normalización de las Telecomunicaciones de la UIT*
- **UMTS** *Universal Mobile Telecommunications System*
- **USB** *Universal Serial Bus*
- **VLR** *Visitor Location Register*
- **VoIP** *Voice Over IP*
- **WAP** *Wireless Application Protocol*
- **WCDMA** *Wideband Code Division Multiple Access*